



Universidad
Carlos III de Madrid

Departamento de Informática

PROYECTO FIN DE CARRERA

Desarrollo de una interfaz distribuida para un escaparate interactivo

Autor: Juan Escudero Vizcaíno

Tutor: Andrea Bellucci

Leganés, junio de 2016

Título: Desarrollo de una interfaz distribuida para un escaparate interactivo

Autor: Juan Escudero Vizcaíno

Director: Andrea Bellucci

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____
de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de
Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Me gustaría mostrar mi agradecimiento y dedicar este proyecto de fin de carrera a:

Mi familia, por haberme dado siempre lo mejor y haberme ayudado a llegar hasta aquí.

Sara, por estar siempre a mi lado, ayudarme y apoyarme.

Mis amigos y compañeros de trabajo, por ofrecerme ayuda siempre que fue necesario.

Andrea, mi tutor, por su ayuda y disposición cada vez que la he necesitado.

Resumen

Actualmente, la tecnología tiene un rol importante en nuestras vidas, permitiéndonos, por ejemplo, comunicarnos con otras personas alrededor del mundo y compartir nuestras opiniones simplemente usando pequeños dispositivos conectados a Internet. Esta manera de expresión ha ido ganando más y más importancia desde la aparición de las redes sociales, donde las personas con intereses similares o conocidas entre ellas están directamente conectadas. Uno de los ejemplos más claros en este caso es Twitter, donde las personas pueden expresarse usando simplemente 140 caracteres. Además, también es más común hoy en día el uso de la tecnología en ámbitos culturales como museos, para dar a sus usuarios una mejor experiencia. Un ejemplo de esto puede ser la disponibilidad de audioguías en museos, la utilización de elementos multimedia para mostrar información sobre obras de arte o la disponibilidad de exposiciones virtuales.

Este proyecto tiene como objetivo facilitar a los usuarios el compartir sus opiniones y experiencias en eventos relacionados con la cultura. Por ejemplo, los usuarios de un museo podrían valorar, comentar e incluso crear un vídeo hablando acerca de cualquier obra de arte presente en dicho museo, además de poder ver las opiniones de otros usuarios acerca de la misma obra. Por otra parte, esta información almacenada podría ser usada por desarrolladores, haciendo uso de la API pública de la aplicación, para recoger los datos generados por los propios usuarios y usarlos en futuras investigaciones.

Las diferentes etapas del diseño y desarrollo de la aplicación están descritas en este documento, empezando por la recolección de prerequisites y concluyendo con la implementación y despliegue de la propia aplicación.

Palabras clave: cultura, tecnología, interacción entre usuarios

Abstract

Nowadays, technology has an important role in our daily life, allowing us to, for example, communicate with other people around the world and share our thoughts by just using small devices connected to the Internet. This way of expressing ourselves has been gaining more importance since the creation of social media, where people with similar interests or with real-life connections are directly connected to each other. One of the most clear examples of this is Twitter, where people express themselves by just using 140 characters. Also, it is now common for cultural events such as museums to use technology in order to give their users a richer experience. This technology usage can go from the existence of audio guides and the usage of multimedia for showing extra information of different pieces of art to the availability of virtual exhibitions.

This project aims to provide a way of letting the users share their experiences and opinions when attending a cultural-related event. As an example, users could rate, make a comment and even create a video talking about a painting inside a museum, and also be able to view other people's opinions about that same object. Moreover, this information could be used by developers by taking advantage of the application public API, in order to retrieve the stored data generated by the own users and use it for future research.

The different stages of the application design and development are described in this document, starting from the collection of the prerequisites and concluding with the final implementation and deployment.

Keywords: culture, technology, users interaction

Índice general

1. INTRODUCCIÓN Y OBJETIVOS	1
1.1 Objetivos	2
1.2 Contenido de la memoria.....	3
2. ESTADO DEL ARTE.....	6
2.1 Aplicaciones Web	6
2.2 Redes Sociales	8
2.3 Nuevas tecnologías en el ámbito cultural	10
2.4 APIs REST	11
2.5 Mashups (aplicaciones web híbridas)	11
3. ANÁLISIS.....	13
3.1 Requisitos de usuario	13
3.1.1 Usuario final.....	15
3.1.2 Desarrollador	18
3.2 Casos de uso	19
3.2.1 Usuario final.....	19
3.2.2 Desarrollador	24
3.3 Requisitos de software	26
3.3.1 Usuario final.....	27
3.3.2 Desarrollador	32
3.4 Matrices de trazabilidad.....	35
3.4.1 Usuario final.....	35
3.4.2 Desarrollador	38
4. DISEÑO	42
4.1 Contexto del sistema.....	42
4.2 Arquitectura de software	44
4.2.1 Arquitectura del sistema.....	45
4.3 Especificación del diseño de componentes	48
4.3.1 Usuario final.....	49

4.3.2 Desarrollador	51
4.4 Matrices de trazabilidad.....	52
4.4.1 Usuario final.....	52
4.4.2 Desarrollador	53
4.5 Prototipos de la interfaz.....	53
4.5.1 Pantalla principal	54
4.5.2 Pantalla de objeto seleccionado	55
4.5.3 Pantalla de vídeo seleccionado.....	56
4.5.4 Pantalla de vídeo seleccionado, sección de comentarios	57
4.5.5 Pantalla de creación de nuevo comentario.....	58
4.5.6 Pantalla de creación de nuevo vídeo	59
4.6 Diseño de API pública.....	60
4.6.1 Objetos.....	60
4.6.2 Vídeos.....	60
4.6.3 Comentarios.....	61
5. IMPLEMENTACIÓN	63
5.1 Tecnologías seleccionadas	63
5.2 Servidor	64
5.2.1 Carpeta “node_modules”.....	65
5.2.2 Archivo “package.json”	65
5.2.3 Archivo “app.js”	65
5.2.4 Archivo “config.js”.....	66
5.2.5 Carpeta “models”	66
5.2.6 Carpeta “views”	66
5.2.7 Carpeta “controllers”	67
5.2.8 Carpeta “db”	67
5.2.9 Archivo “README.md”.....	68
5.3 Cliente	68
5.4 Aspecto final de la aplicación	69
5.4.1 Pantalla principal	69
5.4.2 Pantalla de objeto seleccionado	70
5.4.3 Pantalla de vídeo seleccionado.....	70
5.4.4 Pantalla de vídeo seleccionado, sección de comentarios	71
5.4.5 Pantalla de creación de nuevo comentario.....	71
5.4.6 Pantalla de creación de nuevo vídeo	72
5.4.7 Mensaje de éxito tras crear un comentario.....	72
5.4.8 Mensaje de éxito tras crear un vídeo.....	73
5.5 Pruebas realizadas.....	73
5.6 Integración con plataformas culturales	75
6. GESTIÓN DEL PROYECTO	79
6.1 Modelo de ciclo de vida del software	79
6.2 Planificación	80
6.3 Presupuesto.....	82
6.3.1 Personal.....	82
6.3.2 Material.....	82
6.3.3 Resumen de costes.....	83
7. CONCLUSIONES Y TRABAJO FUTURO	85
7.1 Conclusiones.....	85
7.2 Líneas futuras de mejora.....	86

8. REFERENCIAS	89
9. PROJECT SUMMARY.....	92
Abstract	92
Introduction and objectives	93
<i>Objectives</i>	94
<i>Document content</i>	95
Testing	96
Integrating with cultural platforms	98
Conclusions.....	100
Future improvement areas	101

Índice de figuras

<i>Ilustración 1. Escaparate interactivo para el cual está diseñada la aplicación, incluyendo el lector de códigos RFID en su interior</i>	<i>3</i>
<i>Ilustración 2. Línea temporal de los principales servicios de redes sociales</i>	<i>8</i>
<i>Ilustración 3. Crecimiento de las redes sociales, 2006-2012.....</i>	<i>9</i>
<i>Ilustración 4. Descripción gráfica de casos de uso del usuario final</i>	<i>19</i>
<i>Ilustración 5. Descripción gráfica de casos de uso del desarrollador.....</i>	<i>24</i>
<i>Ilustración 6. Contexto de la aplicación</i>	<i>43</i>
<i>Ilustración 7. Arquitectura Modelo Vista Controlador.....</i>	<i>44</i>
<i>Ilustración 8. Diagrama de componentes (usuario final).....</i>	<i>45</i>
<i>Ilustración 9. Diagrama de componentes (desarrollador).....</i>	<i>47</i>
<i>Ilustración 10. Prototipo de bajo nivel para la pantalla principal.....</i>	<i>54</i>
<i>Ilustración 11. Prototipo de bajo nivel para la pantalla de objeto seleccionado</i>	<i>55</i>
<i>Ilustración 12. Prototipo de bajo nivel para la pantalla de vídeo seleccionado.....</i>	<i>56</i>
<i>Ilustración 13. Prototipo de bajo nivel para la sección de comentarios, en la pantalla de vídeo seleccionado</i>	<i>57</i>
<i>Ilustración 14. Prototipo de bajo nivel para la pantalla de creación de nuevo comentario</i>	<i>58</i>
<i>Ilustración 15. Prototipo de bajo nivel para la pantalla de creación de nuevo vídeo</i>	<i>59</i>
<i>Ilustración 16. Crecimiento de uso e interés de las tecnologías seleccionadas</i>	<i>64</i>
<i>Ilustración 17. Estructura interna del servidor.....</i>	<i>65</i>
<i>Ilustración 18. Estructura de carpetas de la parte del cliente</i>	<i>68</i>
<i>Ilustración 19. Aspecto final de la pantalla principal.....</i>	<i>69</i>
<i>Ilustración 20. Aspecto final de la pantalla de objeto seleccionado</i>	<i>70</i>
<i>Ilustración 21. Aspecto final de la pantalla de vídeo seleccionado.....</i>	<i>70</i>
<i>Ilustración 22. Aspecto final de la pantalla de vídeo seleccionado, sección de comentarios</i>	<i>71</i>
<i>Ilustración 23. Aspecto final de la pantalla de creación de nuevo comentario</i>	<i>71</i>
<i>Ilustración 24. Aspecto final de la pantalla de creación de nuevo vídeo</i>	<i>72</i>
<i>Ilustración 25. Aspecto final del mensaje de éxito tras crear un comentario</i>	<i>72</i>
<i>Ilustración 26. Aspecto final del mensaje de éxito tras crear un vídeo.....</i>	<i>73</i>
<i>Ilustración 27. Entorno de la exhibición (salón principal de la Casa del Lector, situada en el centro de creación contemporánea Matadero de Madrid)</i>	<i>74</i>
<i>Ilustración 28. Usuarios finales usando la aplicación</i>	<i>74</i>
<i>Ilustración 29. Escaparate interactivo mostrando la aplicación en ejecución</i>	<i>75</i>

ÍNDICE DE FIGURAS

<i>Ilustración 30. Pantalla principal del proyecto meSch (mesch-project.eu)</i>	76
<i>Ilustración 31. Pantalla de búsqueda del proyecto meSch</i>	77
<i>Ilustración 32. Ciclo de vida en cascada</i>	80
<i>Ilustración 33. Diagrama de Gantt</i>	81
<i>Ilustración 34. Fórmula de amortización</i>	83
<i>Ilustración 35. Interactive showcase for which the application is designed, including the RFID code reader inside</i>	95
<i>Ilustración 36. Exhibition environment (main room of Casa del Lector, located in the center of contemporary creation Matadero de Madrid)</i>	97
<i>Ilustración 37. End-users using the application</i>	97
<i>Ilustración 38. Interactive showcase showing the application</i>	98
<i>Ilustración 39. Main Screen of meSch project (mesch-project.eu)</i>	99
<i>Ilustración 40. Search screen within meSch project</i>	100

Índice de tablas

<i>Tabla 1. Requisito de capacidad RUC-01</i>	<i>15</i>
<i>Tabla 2. Requisito de capacidad RUC-02</i>	<i>15</i>
<i>Tabla 3. Requisito de capacidad RUC-03</i>	<i>15</i>
<i>Tabla 4. Requisito de capacidad RUC-04</i>	<i>16</i>
<i>Tabla 5. Requisito de restricción RUR-01</i>	<i>16</i>
<i>Tabla 6. Requisito de restricción RUR-02</i>	<i>16</i>
<i>Tabla 7. Requisito de restricción RUR-03</i>	<i>16</i>
<i>Tabla 8. Requisito de restricción RUR-04</i>	<i>17</i>
<i>Tabla 9. Requisito de restricción RUR-05</i>	<i>17</i>
<i>Tabla 10. Requisito de restricción RUR-06</i>	<i>17</i>
<i>Tabla 11. Requisito de capacidad RUC-05</i>	<i>18</i>
<i>Tabla 12. Requisito de capacidad RUC-06</i>	<i>18</i>
<i>Tabla 13. Requisito de capacidad RUC-07</i>	<i>18</i>
<i>Tabla 14. Requisito de restricción RUR-07</i>	<i>18</i>
<i>Tabla 15. Caso de uso CU-01</i>	<i>21</i>
<i>Tabla 16. Caso de uso CU-02</i>	<i>21</i>
<i>Tabla 17. Caso de uso CU-03</i>	<i>22</i>
<i>Tabla 18. Caso de uso CU-04</i>	<i>22</i>
<i>Tabla 19. Caso de uso CU-05</i>	<i>23</i>
<i>Tabla 20. Caso de uso CU-06</i>	<i>23</i>
<i>Tabla 21. Caso de uso CU-07</i>	<i>24</i>
<i>Tabla 22. Caso de uso CU-08</i>	<i>25</i>
<i>Tabla 23. Caso de uso CU-09</i>	<i>25</i>
<i>Tabla 24. Caso de uso CU-09</i>	<i>26</i>
<i>Tabla 25. Requisito funcional RSF-01</i>	<i>27</i>
<i>Tabla 26. Requisito funcional RSF-02</i>	<i>28</i>
<i>Tabla 27. Requisito funcional RSF-03</i>	<i>28</i>
<i>Tabla 28. Requisito funcional RSF-04</i>	<i>28</i>
<i>Tabla 29. Requisito funcional RSF-05</i>	<i>28</i>
<i>Tabla 30. Requisito funcional RSF-06</i>	<i>29</i>
<i>Tabla 31. Requisito funcional RSF-07</i>	<i>29</i>
<i>Tabla 32. Requisito funcional RSF-08</i>	<i>29</i>

ÍNDICE DE TABLAS

<i>Tabla 33. Requisito funcional RSF-09</i>	29
<i>Tabla 34. Requisito funcional RSF-10</i>	30
<i>Tabla 35. Requisito no funcional RSNF-01</i>	30
<i>Tabla 36. Requisito no funcional RSNF-02</i>	30
<i>Tabla 37. Requisito no funcional RSNF-03</i>	30
<i>Tabla 38. Requisito no funcional RSNF-04</i>	31
<i>Tabla 39. Requisito no funcional RSNF-05</i>	31
<i>Tabla 40. Requisito no funcional RSNF-06</i>	31
<i>Tabla 41. Requisito no funcional RSNF-07</i>	31
<i>Tabla 42. Requisito funcional RSF-11</i>	32
<i>Tabla 43. Requisito funcional RSF-12</i>	32
<i>Tabla 44. Requisito funcional RSF-13</i>	32
<i>Tabla 45. Requisito funcional RSF-14</i>	32
<i>Tabla 46. Requisito funcional RSF-15</i>	33
<i>Tabla 47. Requisito funcional RSF-16</i>	33
<i>Tabla 48. Requisito funcional RSF-17</i>	33
<i>Tabla 49. Requisito funcional RSF-18</i>	33
<i>Tabla 50. Requisito funcional RSF-19</i>	34
<i>Tabla 51. Requisito funcional RSF-20</i>	34
<i>Tabla 52. Requisito no funcional RSNF-08</i>	34
<i>Tabla 53. Requisito no funcional RSNF-09</i>	34
<i>Tabla 54. Matriz de trazabilidad entre requisitos de usuario y casos de uso (usuario final)</i>	35
<i>Tabla 55. Matriz de trazabilidad entre requisitos de usuario requisitos de software funcionales (usuario final)</i>	36
<i>Tabla 56. Matriz de trazabilidad entre requisitos de usuario requisitos de software no funcionales (usuario final)</i>	37
<i>Tabla 57. Matriz de trazabilidad entre requisitos de usuario y casos de uso (desarrollador)</i>	38
<i>Tabla 58. Matriz de trazabilidad entre requisitos de usuario y requisitos de software funcionales (desarrollador)</i>	39
<i>Tabla 59. Matriz de trazabilidad entre requisitos de usuario y requisitos de software no funcionales (desarrollador)</i>	40
<i>Tabla 60. Componente CO-01</i>	49
<i>Tabla 61. Componente CO-02</i>	49
<i>Tabla 62. Componente CO-03</i>	50
<i>Tabla 63. Componente CO-04</i>	50
<i>Tabla 64. Componente CO-05</i>	50
<i>Tabla 65. Componente CO-06</i>	50
<i>Tabla 66. Componente CO-07</i>	51
<i>Tabla 67. Componente CO-08</i>	51
<i>Tabla 68. Componente CO-08</i>	51
<i>Tabla 69. Componente CO-09</i>	51
<i>Tabla 70. Tabla de trazabilidad entre requisitos de software y componentes (usuario final)</i>	52
<i>Tabla 71. Tabla de trazabilidad entre requisitos de software y componentes (desarrollador)</i>	53
<i>Tabla 72. Métodos de recolección de datos sobre objetos reales</i>	60
<i>Tabla 73. Métodos de recolección de datos sobre vídeos</i>	60
<i>Tabla 74. Métodos de recolección de datos sobre comentarios</i>	61
<i>Tabla 75. Costes de personal</i>	82
<i>Tabla 76. Costes de material</i>	83
<i>Tabla 77. Resumen de costes</i>	83

Capítulo 1

Introducción y objetivos

Actualmente, Internet tiene una gran importancia en el ámbito de las aplicaciones, siendo un requisito prácticamente indispensable para el correcto funcionamiento de las mismas en entornos móviles, web y escritorio. Por esta razón, el desarrollo de aplicaciones web ha crecido en la última década, ya que estas se basan plenamente en las capacidades de un navegador web, facilitando la integración con la red [\[1\]](#).

Dentro del ámbito de las aplicaciones web, las redes sociales tienen especial importancia debido a la gran cantidad de usuarios activos que poseen [\[2\]](#), los cuales dan un uso diario a las mismas, generando grandes cantidades de información. Esta información puede ser de utilidad en distintas situaciones, ajenas a la aplicación. Un ejemplo de esto es el uso de los datos almacenados en Twitter, donde se puede utilizar la información proporcionada por los usuarios en tiempo real para llevar un seguimiento de situaciones de emergencia [\[3\]](#).

Teniendo en cuenta esta versatilidad de las aplicaciones web y la generación de información que conllevan, este proyecto tiene como principal objetivo el desarrollo de una aplicación para eventos de carácter cultural, basada en un enfoque similar al de las redes sociales, proporcionando a los usuarios un modo de interactuar con otros visitantes de dichos eventos. La información generada mediante el uso de esta aplicación será accesible para desarrolladores e investigadores, posibilitando futuras investigaciones y aplicaciones basadas en ella.

Con respecto a la tecnología actualmente usada en eventos de ámbito cultural, esta está enfocada hacia los tours y catálogos virtuales de museos, los cuales consisten en una representación virtual del contenido que ofrece dicho evento. Estos tours y catálogos son accesibles comúnmente por Internet, lo cual aporta tanto ventajas como inconvenientes, los cuales serán explicados más adelante.

En la actualidad, el conservador de bienes culturales o el encargado del museo es quien decide que información estará presente en una exposición. Con este trabajo se propone habilitar a los visitantes para que puedan participar de forma activa en la creación de narrativas alternativas a las propuestas por los conservadores [\[4\]](#).

La principal motivación de este proyecto nace de la falta herramientas que permitan a los usuarios interactuar entre ellos en un evento en el cual todos comparten un factor común. Es interesante recalcar que la tecnología en este ámbito es usada actualmente desde otro enfoque, mejorando la experiencia que el propio evento pueda proporcionar a los usuarios, pero no favoreciendo la comunicación entre ellos. Asimismo, este proyecto presenta también una herramienta para desarrolladores, permitiendo el acceso a los datos generados por los usuarios. De esta forma, el uso de la aplicación a desarrollar repercutirá en el crecimiento de la misma, aportando un catálogo virtual extenso y rico en contenido variado.

La decisión de usar tecnologías web en este proyecto reside en la facilidad que proporcionan a la hora de desarrollar una aplicación multiplataforma y su integración con Internet, debido a la intención de permitir el acceso a la información almacenada [\[5\]](#).

1.1 Objetivos

El presente proyecto se centra en el estudio y desarrollo de un prototipo de aplicación web que ofrezca a los usuarios de eventos culturales la posibilidad de expresar, comunicar y valorar opiniones de otros usuarios del mismo evento.

Debido a lo anteriormente explicado, se pretende diseñar una aplicación web dirigida especialmente a dos tipos de usuarios diferenciados:

- Usuarios finales: serán aquellos usuarios que utilicen la aplicación con el propósito de interactuar con las obras (compartir experiencias relacionadas con dichas obras, valorarlas y comentarlas) y otros usuarios (mediante la comunicación basada en vídeo y texto). Como consecuencia, el uso de la aplicación por parte de estos usuarios implicará la generación de datos y su posterior almacenamiento.
- Desarrolladores: los cuales accederán a los datos almacenados por parte de la aplicación, con el fin de utilizarlos en proyectos ajenos al actual.

Debido a que el proyecto tendrá los dos tipos de usuarios anteriormente mencionados, es importante señalar que la interacción con la aplicación será distinta en ambos casos:

- En el caso de los usuarios finales, la aplicación será utilizada mediante un escaparate interactivo y un lector de códigos RFID [\[6\]](#), lo cual facilitará la interacción con la obra elegida, al mismo tiempo que proporcionará cierto grado de inmersión con la misma.



Ilustración 1. Escaparate interactivo para el cual está diseñada la aplicación, incluyendo el lector de códigos RFID en su interior

- En el caso de los desarrolladores, el acceso a los datos almacenados de la aplicación podrá realizarse desde cualquier entorno que soporte el protocolo HTTP. Se diseñará e implementará una API pública para la comunicación con la capa de servidor de la aplicación desde un entorno externo.

1.2 Contenido de la memoria

El presente documento se encuentra dividido en varios capítulos a lo largo de los cuales se recoge toda la información relativa al desarrollo del proyecto. Estos capítulos son:

- Introducción: se tratará de dar una visión global del proyecto, qué objetivos se han establecido y cual será el contenido del resto del documento.

CAPÍTULO 1: INTRODUCCIÓN y objetivos

- Estado del arte: se hablará sobre algunas generalidades de las tecnologías web y cómo éstas han sido clave en el desarrollo de aplicaciones durante los últimos años, además de cómo la tecnología y el ámbito cultural se han unido con el fin de enriquecer la experiencia de los usuarios.
- Análisis: se cubrirán las funcionalidades de la aplicación, teniendo en cuenta la funcionalidad que el usuario espera de ella. Posteriormente se detallarán los casos de uso, donde se podrá ver con claridad las operaciones que los usuarios podrán llevar a cabo con el uso de la aplicación.
- Diseño: el contexto del sistema y la arquitectura de la aplicación serán explicadas, al igual que la matriz de trazabilidad del proyecto.
- Implementación y desarrollo: se hablará sobre las tecnologías utilizadas en el desarrollo de la aplicación.
- Gestión del proyecto: en este capítulo se describe y justifica el modelo de ciclo de vida elegido para el desarrollo del proyecto. Posteriormente se presenta la planificación seguida, así como los costes asociados a la elaboración del proyecto.
- Conclusiones: en este capítulo se sintetizan los principales resultados del proyecto y se ofrecen posibles líneas de mejora para un futuro.
- Referencias: en este capítulo se listan los recursos bibliográficos consultados durante la elaboración del proyecto.

Capítulo 2

Estado del arte

Previamente al desarrollo de la aplicación, es necesario realizar una aproximación al concepto de aplicaciones web. Asimismo, es también interesante comprender qué son las APIs REST y cómo pueden ayudar al desarrollo de este proyecto. Por último, se efectúa una aproximación a las tecnologías mas apropiadas para alcanzar las metas.

2.1 Aplicaciones Web

Antiguamente, el uso de la Web implicaba sólo documentos estáticos que, unidos entre ellos mediante vínculos, otorgaban al usuario una sensación de interactividad. A pesar de esto, cualquier acción por parte del usuario con el documento implicaría un envío de información al servidor y una respuesta por parte del mismo, por lo que el documento sería cargado de nuevo [\[7\]](#). La evolución de este primer punto hasta la situación actual se puede describir con varios hitos que significaron cambios de vital importancia para el desarrollo de las actuales aplicaciones web:

- En 1995 se introduce el lenguaje de scripting en el lado del cliente, JavaScript, por parte de Netscape. Con él, los programadores podían añadir elementos dinámicos a la interfaz de usuario del documento cargado.

- En 1996, Macromedia introduce un reproductor de animaciones vectoriales llamado Flash, que podía añadirse como plug-in para reproducir animaciones en páginas web. Los desarrolladores podían programar interacciones en el lado del cliente sin necesidad de una comunicación con el servidor.
- En 1999 aparece el concepto de “aplicación web” en la especificación 2.2 de Java Servlet.
- En 2005 se acuña el término AJAX (Asynchronous Javascript And XML). Es una técnica en la que un script de JavaScript en el lado del cliente es capaz de contactar con el servidor para enviar y/o recibir información sin tener que cargar la página por completo de nuevo. Aplicaciones como Gmail comienzan a mejorar en términos de interactividad debido a esta técnica.
- En 2011 se finaliza el desarrollo de HTML5, incorporando capacidades de gráficos y multimedia sin necesidad de plug-ins en el lado del cliente. Se incluyen también APIs nuevas del DOM (Document Object Model) y WebGL, permitiendo hacer aplicaciones de Internet enriquecidas (RIA en inglés).

Actualmente, existe un gran número de aplicaciones web, cubriendo un gran abanico de necesidades, desde la edición fotográfica (Pixlr) y el aprendizaje (Udemy) hasta el trabajo en equipo de prototipado de aplicaciones (Invision). Debido a esto, compañías como Google y Mozilla han centrado su atención en el desarrollo de Sistemas Operativos cuyas aplicaciones están basadas únicamente en tecnologías web. Estos proyectos son Chrome OS y Firefox OS respectivamente, los cuales no han tenido la aceptación esperada debido al principal problema de las aplicaciones web: el requisito de conexión a Internet.

Es por este principal requisito por lo que las aplicaciones web están teniendo una integración lenta en la sociedad, pero existen proyectos para paliar esta brecha entre las aplicaciones de escritorio y las aplicaciones web, haciendo posible la “traducción” de las aplicaciones web a aplicaciones nativas, como Electron.

Es interesante observar, teniendo en cuenta lo anteriormente mencionado, que no existe ninguna solución perfecta en cuanto a aplicaciones se refiere. Las aplicaciones web tienen como principales ventajas la facilidad de desarrollo y su uso en distintas plataformas sin necesidad de adaptarse a las mismas, y como principales inconvenientes el requisito de conexión a Internet y las limitaciones de los navegadores web. En cambio, las aplicaciones nativas tienen la ventaja de poder usar de manera más eficiente el hardware del dispositivo, pero es necesario adaptar la misma aplicación a las distintas plataformas donde se requiera instalar, necesitando usar distintas tecnologías en el desarrollo de la misma aplicación para este fin.

2.2 Redes Sociales

El primer sitio web al que nos podemos referir como red social es SixDegrees.com, lanzado en 1997, definido como una herramienta para ayudar a la gente a conectar con otras personas y poder comunicarse entre ellos mediante mensajes. A pesar de tener un éxito inicial atrayendo a millones de usuarios, no consiguió convertirse en un negocio sostenible, cerrándose el servicio en el año 2000 [\[8\]](#).

Tras SixDegrees, con el tiempo aparecieron nuevos servicios con más éxito, entre los que destacan Friendster (2002), MySpace (2003), LinkedIn (2003), XING (2003) y Facebook (2004/2006) [\[9\]](#). En la siguiente ilustración se pueden apreciar los principales servicios de redes sociales fundados desde la aparición de SixDegrees.com:

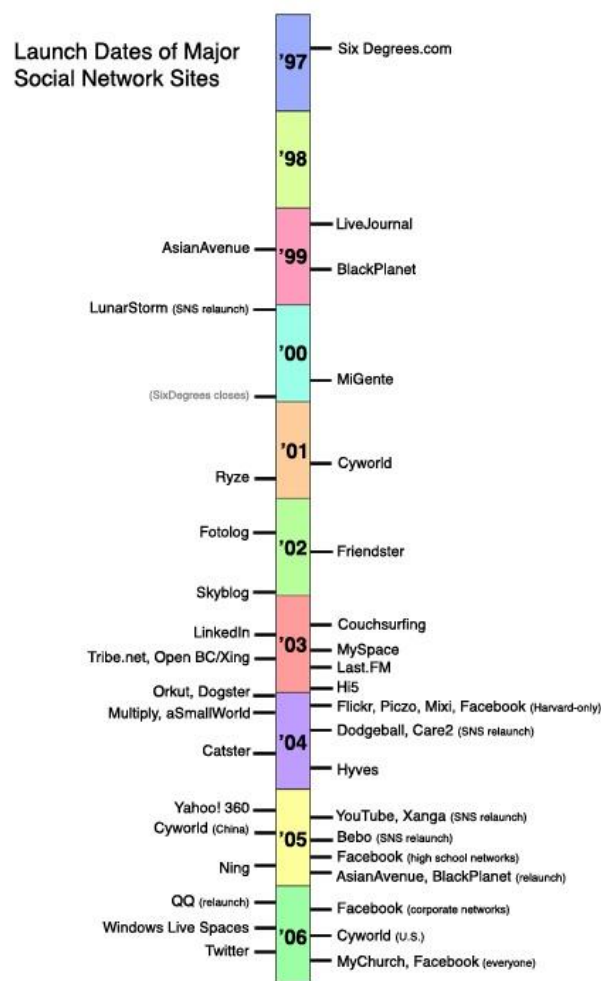
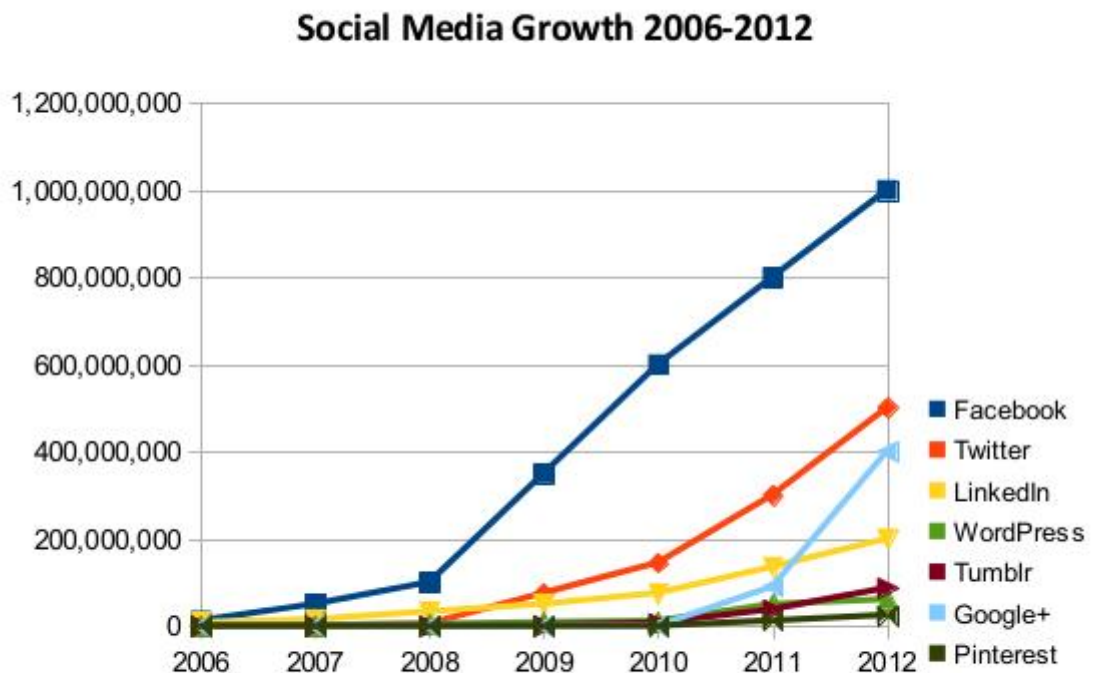


Ilustración 2. Línea temporal de los principales servicios de redes sociales

Actualmente, las principales redes sociales han ido creciendo en número de usuarios, llegando a cifras millonarias en el caso de Facebook.



<http://www.dstevenwhite.com>

Ilustración 3. Crecimiento de las redes sociales, 2006-2012

Se puede apreciar que la red social Twitter (creada en 2006 por Jack Dorsey) está dentro del grupo de redes sociales mas utilizadas. Debido a esto y a su sencillez, el presente proyecto toma como principal inspiración a esta red social.

La principal funcionalidad que brinda Twitter a sus usuarios es la de comunicarse mediante el uso de mensajes de texto plano de corta longitud, teniendo un máximo de 140 caracteres [10]. Además de esto, los mensajes pueden incluir imágenes o vídeos, enriqueciendo así la comunicación entre sus usuarios. Actualmente, el principal uso que recibe Twitter es el de compartir noticias en tiempo real y la comunicación entre usuarios (principalmente la expresión personal).

Esta forma de comunicación es la elegida para este proyecto debido a su facilidad de aprendizaje y uso, permitiendo a los usuarios compartir sus opiniones y experiencias para que otros usuarios puedan recibir esa información e interactuar con la misma. De esta forma, estaríamos creando un medio de comunicación similar a una red social, centrado en eventos culturales.

2.3 Nuevas tecnologías en el ámbito cultural

Actualmente, la unión entre tecnología y patrimonio cultural está en continua evolución. Hasta hace una década, la tecnología en eventos del ámbito cultural era usada para enriquecer la visita mediante dispositivos como las audioguías, o para facilitar información sobre las obras a los visitantes, mediante el uso de elementos multimedia (texto, vídeo y audio).

El uso de las nuevas tecnologías en el ámbito cultural ha evolucionado con el paso de los años, dando lugar a la creación de museos virtuales [\[11\]](#). Estos museos virtuales se caracterizan por el uso de medios informáticos para mostrar, preservar, estudiar, reconstruir y divulgar el patrimonio material o inmaterial de la humanidad. Las principales ventajas de este tipo de museos frente a los convencionales museos físicos:

- Permiten el acceso gratuito de cualquier persona en todo el mundo y lugar a representaciones de obras de arte reales
- Permiten el acceso a colecciones y piezas que no se encuentran expuestas en las salas del museo físico
- No están sujetos a limitaciones de espacio y tiempo, existentes en los museos físicos, siendo posible la organización simultánea de obras según distintos criterios y contextos
- Permiten una interacción más implicada por parte del usuario, existiendo conferencias y foros de discusión para este fin

Es importante destacar que, a pesar de tener numerosas ventajas, los museos virtuales requieren ciertos conocimientos para el correcto uso y el acceso a todas las posibilidades que ofrecen dichos museos. Por esta razón, este tipo de museos no debe únicamente confiar en los conocimientos informáticos de la población, sino que también deben intentar simplificar el uso y acceso a estas plataformas.

Actualmente, museos físicos como el Museo Nacional de Historia Natural del Instituto Smithsonian, en Washington D.C., ofrecen de forma gratuita acceso a tours virtuales via web, simulando una visita guiada por el museo físico, basados tanto en fotografías panorámicas de 360º como en entornos basados en modelado 3D.

En el ámbito de la investigación y el desarrollo, gracias a las nuevas posibilidades que ofrece este tipo de museos, cada vez se abren más vías de investigación, fundamentadas principalmente en el libre acceso a las bases de datos completas de cada museo. Estas bases de datos no constituyen en sí mismas museos virtuales, pero almacenan la información necesaria para que desarrolladores e investigadores puedan hacer uso de la misma con el fin de llevar a cabo proyectos basados en los datos que proporcionan este tipo de museos.

2.4 APIs REST

El término API (del inglés: Application Programming Interface) hace referencia a una capa de abstracción en cuanto a software se refiere. La principal utilidad que tiene una API es la de proporcionar funcionalidad que pueda ser accedida por otro software [\[12\]](#). En el caso de la red social Twitter, su API permite a desarrolladores poder recoger tweets almacenados en las bases de datos de la aplicación y poder mostrarlos en sitios web externos a Twitter, como puede ser un blog de carácter personal.

El término REST (Representational State Transfer) se usa para designar un estilo de arquitectura de software, que utiliza directamente el protocolo HTTP para obtener datos o indicar operaciones sobre esos datos [\[13\]](#). REST nos permite crear servicios y aplicaciones que pueden ser usadas por cualquier dispositivo o cliente que pueda hacer uso del estándar HTTP. Por esto, REST es el tipo de arquitectura mas natural para crear APIs para servicios orientados a Internet.

Mediante el uso de APIs REST diseñadas para esta aplicación, este proyecto brindará la posibilidad a desarrolladores de acceder a la información generada por los usuarios y almacenada en la base de datos. De esta forma, los datos existentes podrán ser útiles no solo de cara a este proyecto, sino también para futuras investigaciones y estudios relacionados con la tecnología y la herencia cultural.

2.5 Mashups (aplicaciones web híbridas)

En el caso del desarrollo de una aplicación basada en los datos generados por la aplicación, se estaría haciendo referencia a un concepto conocido como mashup [\[14\]](#). Una mashup, o aplicación web híbrida, es aquella que utiliza contenido procedente de más de una fuente, con el fin de crear un único servicio, mostrado en una única interfaz gráfica.

Un ejemplo de aplicación híbrida sería la combinación de direcciones y fotografías guardadas en una base de datos junto con un mapa de Google Maps, con el fin de asociar dichas fotografías a las direcciones y mostrarlas dentro de un mapa en la aplicación web.

Teniendo en cuenta lo anteriormente dicho, este proyecto servirá de fuente de información para aquellos desarrolladores que la requieran para crear una aplicación web híbrida, facilitando dicha información mediante la API pública asociada a este proyecto.

Capítulo 3

Análisis

Una vez analizado el estado del arte, el siguiente objetivo es elaborar el catálogo de requisitos del sistema para lo cual se comienza por la obtención de los requisitos de usuario, paso con el que se obtiene la principal funcionalidad que el usuario espera de la aplicación. Posteriormente se especifican los casos de uso que permiten obtener la percepción del usuario sobre la interacción con el sistema. El paso siguiente es el desglose de la información obtenida en los requisitos de software más específicos y detallados. Finalmente se realizan las matrices de trazabilidad para comprobar la consistencia de los requisitos.

Dado que la aplicación ofrecerá funcionalidades tanto al usuario final como a desarrolladores, el presente capítulo se dividirá en dos secciones para cubrir ambos casos.

3.1 Requisitos de usuario

Mediante estos requisitos se define la funcionalidad de la aplicación con respecto a las necesidades de los usuarios y también se especifican sus restricciones. Se pueden dividir en dos clases:

CAPÍTULO 3: Análisis

- Requisitos de capacidad: aquellos que definen la funcionalidad que la aplicación debe proporcionar.
- Requisitos de restricción: requisitos que especifican limitaciones (de recursos, temporales, etc.) sobre la funcionalidad anterior.

Para la especificación de los diferentes requisitos se utilizan los siguientes campos:

- Identificador: permite reconocer cada requisito de manera unívoca. Este campo cumple con la siguiente nomenclatura: RUC-XX (en el caso de los requisitos de capacidad) o RUR-XX (en el caso de los requisitos de restricción), donde XX corresponde a dos dígitos numéricos. Este identificador no tiene la obligación de guardar relación con el identificador del requisito anterior.
- Nombre: describe de manera breve el objetivo del requisito.
- Descripción: especifica de manera completa en qué consiste el requisito.
- Fuente: determina el origen del requisito.
- Necesidad: especifica el grado de importancia que tiene este requisito para el proyecto, pudiendo tomar uno de los siguientes valores: esencial, deseable u opcional.
- Prioridad: especifica la urgencia con la que el requisito debe cumplirse. Los valores a tomar por este campo son: alta, media o baja.
- Claridad: determina si el requisito es entendible por si solo y de manera correcta. Los valores que puede tomar este campo son: alta, media o baja.
- Estabilidad: especifica si el requisito puede variar a lo largo de la vida del proyecto, pudiendo tomar uno de los siguientes valores: estable o inestable.
- Verificabilidad: determina si el cumplimiento del requisito puede comprobarse. Los valores que puede tomar este campo son: alta, media o baja.

3.1.1 Usuario final

3.1.1.1 Requisitos de capacidad

RUC-01			
Nombre	Interacción con objetos reales		
Descripción	El usuario podrá interactuar con el objeto real deseado, pudiendo manipularlo y recibir información del mismo una vez haya sido colocado sobre el lector de códigos RFID		
Fuente	Analista	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 1. Requisito de capacidad RUC-01

RUC-02			
Nombre	Interacción con vídeos de usuarios		
Descripción	Tras haber seleccionado el objeto real, el usuario podrá interactuar con los vídeos asociados al mismo, pudiendo llevar a cabo las siguientes acciones: <ul style="list-style-type: none"> - Seleccionar un vídeo de la lista - Reproducir el vídeo, aumentando el número de visualizaciones - Valorar el vídeo (positiva o negativamente) 		
Fuente	Analista	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 2. Requisito de capacidad RUC-02

RUC-03			
Nombre	Interacción con comentarios		
Descripción	Cada vídeo tiene asociada una lista de comentarios creados por usuarios. Una vez el vídeo se haya seleccionado, el usuario podrá: <ul style="list-style-type: none"> - Acceder a los comentarios ya existentes - Crear un nuevo comentario, pudiendo escribir un texto dentro del mismo y grabar un nuevo vídeo - Responder a un comentario ya existente 		
Fuente	Analista	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 3. Requisito de capacidad RUC-03

RUC-04			
Nombre	Crear nuevo vídeo		
Descripción	El usuario podrá añadir un nuevo vídeo a la lista de videos relacionados con el objeto seleccionado		
Fuente	Analista	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 4. Requisito de capacidad RUC-04

3.1.1.2 Requisitos de restricción

RUR-01			
Nombre	Idioma		
Descripción	El idioma de la aplicación será el Inglés		
Fuente	Analista	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 5. Requisito de restricción RUR-01

RUR-02			
Nombre	Navegador		
Descripción	La aplicación será compatible con el navegador web Google Chrome		
Fuente	Analista	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 6. Requisito de restricción RUR-02

RUR-03			
Nombre	Formato de vídeo		
Descripción	El formato de todos los videos almacenados deberá ser WebM		
Fuente	Analista	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 7. Requisito de restricción RUR-03

RUR-04			
Nombre	Sistema Operativo		
Descripción	Los Sistemas Operativos soportados serán Windows y OS X		
Fuente	Analista	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 8. Requisito de restricción RUR-04

RUR-05			
Nombre	Lector RFID		
Descripción	Para la identificación de objetos reales, será necesario el uso de un lector de RFIDs conectado a un dispositivo Arduino, además de adhesivos con códigos RFID para su uso en los objetos		
Fuente	Analista	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 9. Requisito de restricción RUR-05

RUR-06			
Nombre	Interfaz		
Descripción	La interfaz será clara y fácil de utilizar, haciendo uso de iconos descriptivos, fuente y controles de tamaño razonable y fácil navegación		
Fuente	Analista	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 10. Requisito de restricción RUR-06

3.1.2 Desarrollador

3.1.2.1 Requisitos de capacidad

RUC-05			
Nombre	Recolección de información sobre los objetos		
Descripción	Recolectar información sobre los objetos reales a los que da soporte la aplicación, mediante el uso de la API		
Fuente	Analista	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 11. Requisito de capacidad RUC-05

RUC-06			
Nombre	Recolección de información sobre los vídeos		
Descripción	Recolectar información sobre los vídeos almacenados en la aplicación, mediante el uso de la API		
Fuente	Analista	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 12. Requisito de capacidad RUC-06

RUC-07			
Nombre	Recolección de información sobre los comentarios		
Descripción	Recolectar información sobre los comentarios almacenados en la aplicación, mediante el uso de la API		
Fuente	Analista	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 13. Requisito de capacidad RUC-07

3.1.2.2 Requisitos de restricción

RUR-07			
Nombre	Formato de los datos recolectados		
Descripción	El formato que la aplicación devolverá una vez hecha la petición de recolección será JSON (JavaScript Object Notation)		
Fuente	Analista	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 14. Requisito de restricción RUR-07

3.2 Casos de uso

Mediante los casos de uso se representa la interacción y comunicación de los usuarios con la aplicación a través de escenarios. Con esta técnica se persigue alcanzar una mayor completitud de requisitos.

Para su especificación, se utilizan dos representaciones: la descripción gráfica de los casos de uso y la descripción textual de los mismos.

3.2.1 Usuario final

3.2.1.1 Descripción gráfica

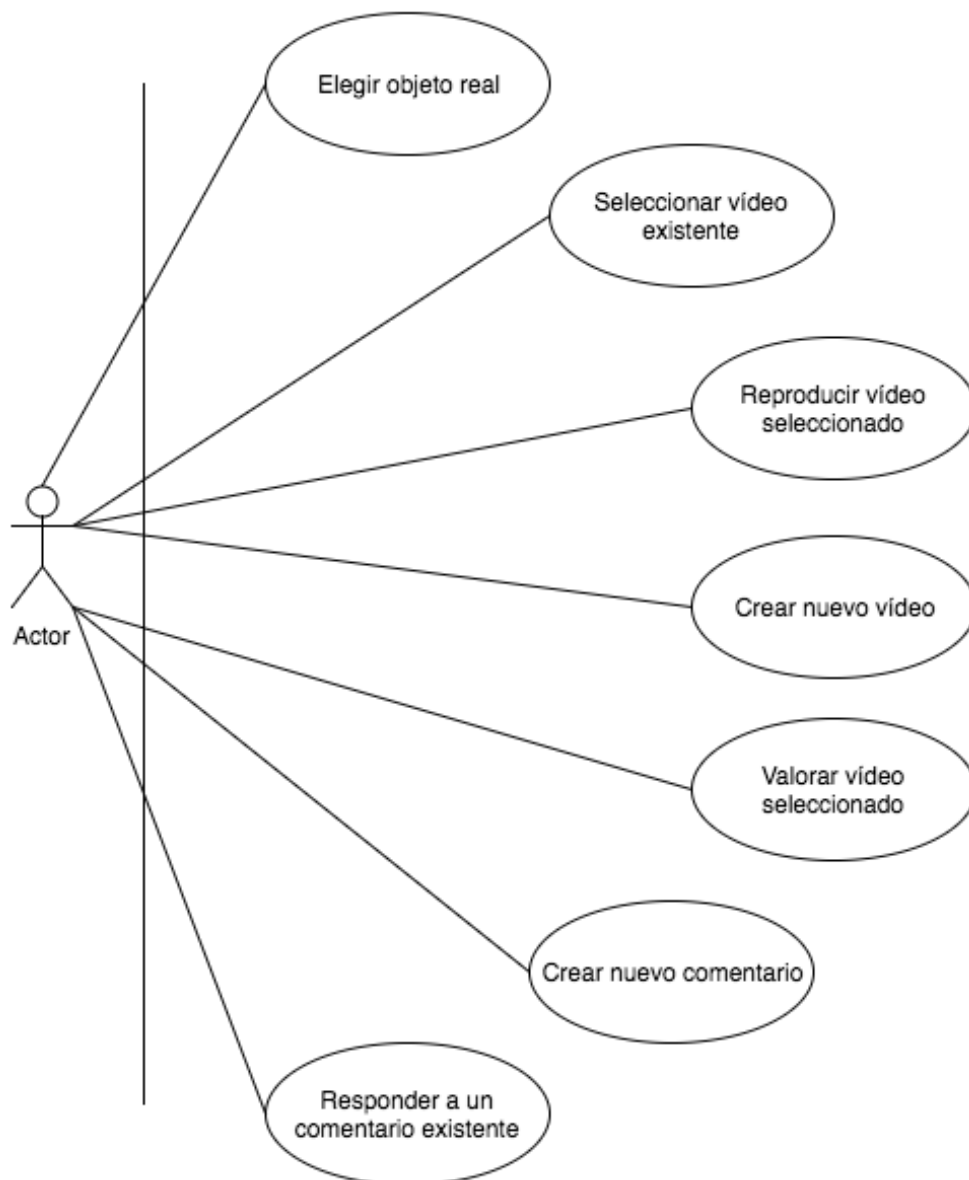


Ilustración 4. Descripción gráfica de casos de uso del usuario final

3.2.1.2 Descripción textual

Para llevar a cabo la descripción textual de los diferentes casos de uso se utilizan los siguientes campos:

- Identificador: permite reconocer cada caso de uso de manera unívoca. Este campo cumple con la siguiente nomenclatura: CU-XX donde XX se corresponde con dígitos numéricos. Este identificador no tiene la obligación de guardar relación con el identificador del caso de uso anterior.
- Nombre: describe de manera breve el objetivo del caso de uso.
- Actores: especifica los sistemas externos que requieren y participan del caso de uso.
- Objetivo: especifica de manera concisa la finalidad del caso de uso.
- Precondiciones: especifica las condiciones que deben darse en el sistema para poder comenzar a ejecutar el caso de uso.
- Postcondiciones: especifica las condiciones que se darán en el sistema tras finalizarse la ejecución del caso de uso.
- Escenario básico: especifica la secuencia de pasos que han de llevarse a cabo para ejecutar correctamente el caso de uso.
- Escenario alternativo: especifica alternativas al escenario básico.

CU-01	
Nombre	Elegir objeto real
Actores	Usuario
Objetivo	Seleccionar un objeto real para su posterior identificación mediante el uso del lector de RFIDs asociado a la aplicación
Precondiciones	Existencia del objeto en la base de datos de la aplicación (asociado a su código RFID)
Postcondiciones	El objeto es identificado por el lector de RFIDs, mostrándose información sobre el mismo en la pantalla principal de la aplicación
Escenario básico	<ol style="list-style-type: none"> 1. El usuario selecciona el objeto real y lo coloca sobre el lector de RFIDs 2. El lector de RFIDs le proporciona a la aplicación el código identificado 3. La aplicación realiza una búsqueda en la base de datos para recolectar información sobre el objeto seleccionado 4. Se muestra en la pantalla principal de la aplicación el título del objeto seleccionado, un pequeño resumen del mismo y los vídeos asociados
Escenario alternativo	Ninguno

Tabla 15. Caso de uso CU-01

CU-02	
Nombre	Seleccionar vídeo existente
Actores	Usuario
Objetivo	Seleccionar un vídeo existente relacionado con el objeto previamente seleccionado, con el fin de reproducirlo, valorarlo y crear contenido relacionado con el mismo
Precondiciones	Objeto real previamente seleccionado e identificado por el lector de RFIDs
Postcondiciones	El vídeo es mostrado en la pantalla principal de la aplicación, permitiendo al usuario llevar a cabo distintas acciones relacionadas con el mismo
Escenario básico	<ol style="list-style-type: none"> 1. Una vez que la lista de vídeos relacionados con el objeto seleccionado es mostrada, el usuario selecciona un elemento de la misma 2. La información mostrada en la pantalla cambia, dejando de mostrar información relacionada con el objeto para dar paso a la información relacionada con el vídeo seleccionado
Escenario alternativo	Ninguno

Tabla 16. Caso de uso CU-02

CU-03	
Nombre	Reproducir vídeo seleccionado
Actores	Usuario
Objetivo	Reproducir el contenido del vídeo seleccionado
Precondiciones	Vídeo previamente seleccionado de la lista de vídeos relacionados con el objeto real
Postcondiciones	El vídeo es reproducido en la pantalla principal de la aplicación, y el número de reproducciones de ese mismo vídeo aumenta una unidad
Escenario básico	1. Después de seleccionar el vídeo relacionado con el objeto, el usuario pulsa sobre el botón de reproducción, visible en la pantalla principal de la aplicación
Escenario alternativo	Ninguno

Tabla 17. Caso de uso CU-03

CU-04	
Nombre	Valorar vídeo seleccionado
Actores	Usuario
Objetivo	Valorar positiva o negativamente el contenido del vídeo seleccionado
Precondiciones	Vídeo previamente seleccionado de la lista de vídeos relacionados con el objeto real
Postcondiciones	La valoración del vídeo es almacenada y mostrada en la pantalla principal del vídeo
Escenario básico	1. Después de seleccionar el vídeo relacionado con el objeto, el usuario pulsa sobre uno de los botones de valoración
Escenario alternativo	Ninguno

Tabla 18. Caso de uso CU-04

CU-05	
Nombre	Crear nuevo comentario sobre el vídeo seleccionado
Actores	Usuario
Objetivo	Crear un nuevo comentario relacionado con el vídeo seleccionado
Precondiciones	Vídeo previamente seleccionado de la lista de vídeos relacionados con el objeto real
Postcondiciones	El comentario es almacenado y añadido a la lista de comentarios relacionados con el vídeo seleccionado
Escenario básico	<ol style="list-style-type: none"> 1. Después de seleccionar el vídeo relacionado con el objeto, el usuario pulsa sobre el botón de creación de nuevo comentario, habilitando la pantalla para este fin 2. El usuario rellena los campos de texto necesarios para crear el nuevo comentario 3. Se pulsa sobre el botón de guardado de comentario 4. La aplicación muestra nuevamente el vídeo seleccionado y la lista de comentarios actualizada
Escenario alternativo	2a. El usuario no rellena los campos de texto. En su lugar, crea un nuevo vídeo como contenido del comentario
Escenario alternativo	2b. El usuario rellena los campos de texto necesarios para crear el nuevo comentario, además de crear un nuevo vídeo como contenido del comentario

Tabla 19. Caso de uso CU-05

CU-06	
Nombre	Responder a un comentario existente
Actores	Usuario
Objetivo	Crear un nuevo comentario en respuesta a un comentario ya existente
Precondiciones	Existencia de, al menos, un comentario sobre el vídeo seleccionado
Postcondiciones	El comentario es almacenado y añadido a la lista de respuestas de un comentario existente
Escenario básico	<ol style="list-style-type: none"> 1. Después de seleccionar el vídeo relacionado con el objeto, el usuario pulsa sobre el botón de creación de respuesta de un comentario ya existente, habilitando la pantalla para este fin 2. El usuario rellena los campos de texto necesarios para crear el nuevo comentario 3. Se pulsa sobre el botón de guardado de comentario 4. La aplicación muestra nuevamente el vídeo seleccionado y la lista de comentarios actualizada
Escenario alternativo	2a. El usuario no rellena los campos de texto. En su lugar, crea un nuevo vídeo como contenido del comentario
Escenario alternativo	2b. El usuario rellena los campos de texto necesarios para crear el nuevo comentario, además de crear un nuevo vídeo como contenido del comentario

Tabla 20. Caso de uso CU-06

CU-07	
Nombre	Crear nuevo vídeo
Actores	Usuario
Objetivo	Crear un nuevo vídeo relacionado con el objeto seleccionado
Precondiciones	Objeto real seleccionado
Postcondiciones	Un nuevo vídeo será almacenado y añadido a la lista de vídeos relacionados con el objeto seleccionado
Escenario básico	<ol style="list-style-type: none"> 1. Tras seleccionar el objeto real, la aplicación mostrará información relacionada con el objeto y la lista de vídeos relacionados con el mismo 2. El usuario pulsa el botón de creación de nuevo vídeo 3. La aplicación muestra la pantalla de creación de nuevo vídeo 4. El usuario presiona el botón de grabación 5. Tras grabar el vídeo, el usuario presiona el botón de guardar el vídeo grabado
Escenario alternativo	Ninguno

Tabla 21. Caso de uso CU-07

3.2.2 Desarrollador

3.2.2.1 Descripción gráfica

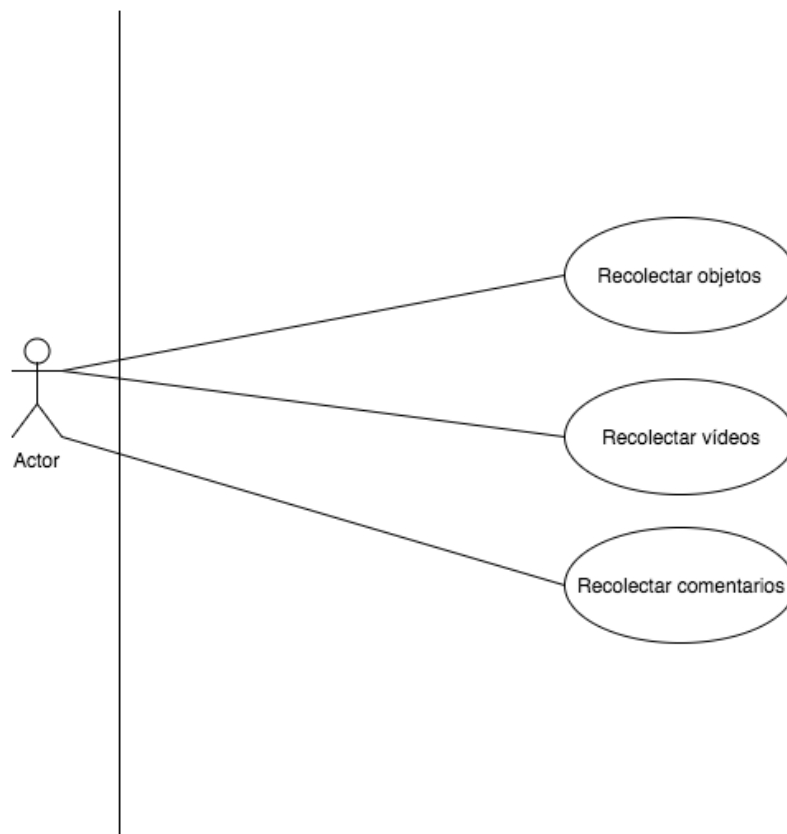


Ilustración 5. Descripción gráfica de casos de uso del desarrollador

3.2.2.2 Descripción textual

CU-08	
Nombre	Recolectar objetos
Actores	Desarrollador
Objetivo	Recolectar información sobre los objetos reales a los que la aplicación da soporte
Precondiciones	Existencia de los objetos reales en la base de datos de la aplicación
Postcondiciones	Un objeto JSON será devuelto al desarrollador, conteniendo información sobre los objetos deseados
Escenario básico	<ol style="list-style-type: none"> 1. Se realiza una petición al servidor de la aplicación, especificando que se desea recibir información sobre todos los objetos 2. Se recibe un objeto JSON con información sobre todos los objetos
Escenario alternativo	<ol style="list-style-type: none"> 1a. Se realiza una petición al servidor de la aplicación, especificando el código RFID del objeto deseado 2a. Se recibe un objeto JSON con información sobre el objeto con ese RFID asociado
Escenario alternativo	<ol style="list-style-type: none"> 1b. Se realiza una petición al servidor de la aplicación, especificando el nombre del objeto deseado 2b. Se recibe un objeto JSON con información sobre todos los objetos que coincidan con el nombre recibido

Tabla 22. Caso de uso CU-08

CU-09	
Nombre	Recolectar vídeos
Actores	Desarrollador
Objetivo	Recolectar información sobre los vídeos asociados a objetos reales
Precondiciones	Existencia de vídeos asociados a objetos reales en la base de datos de la aplicación
Postcondiciones	Un objeto JSON será devuelto al desarrollador, conteniendo información sobre los vídeos deseados
Escenario básico	<ol style="list-style-type: none"> 1. Se realiza una petición al servidor de la aplicación, especificando que se desea recibir información sobre todos los vídeos 2. Se recibe un objeto JSON con información sobre todos los vídeos
Escenario alternativo	<ol style="list-style-type: none"> 1a. Se realiza una petición al servidor de la aplicación, especificando el código RFID del objeto sobre el cual tratan los vídeos 2a. Se recibe un objeto JSON con información sobre todos los vídeos asociados a ese objeto
Escenario alternativo	<ol style="list-style-type: none"> 1b. Se realiza una petición al servidor de la aplicación, especificando el nombre del objeto sobre el cual tratan los vídeos 2b. Se recibe un objeto JSON con información sobre todos los vídeos asociados a ese objeto

Tabla 23. Caso de uso CU-09

CU-09	
Nombre	Recolectar comentarios
Actores	Desarrollador
Objetivo	Recolectar información sobre los comentarios asociados vídeos almacenados en la aplicación
Precondiciones	Existencia de vídeos asociados a objetos reales en la base de datos de la aplicación
Postcondiciones	Un objeto JSON será devuelto al desarrollador, conteniendo información sobre los comentarios deseados
Escenario básico	<ol style="list-style-type: none"> 1. Se realiza una petición al servidor de la aplicación, especificando que se desea recibir información sobre todos los comentarios 2. Se recibe un objeto JSON con información sobre todos los comentarios
Escenario alternativo	<ol style="list-style-type: none"> 1a. Se realiza una petición al servidor de la aplicación, especificando el código RFID del objeto sobre el cual tratan los comentarios 2a. Se recibe un objeto JSON con información sobre todos los comentarios asociados a ese objeto
Escenario alternativo	<ol style="list-style-type: none"> 1b. Se realiza una petición al servidor de la aplicación, especificando el nombre del objeto sobre el cual tratan los comentarios 2b. Se recibe un objeto JSON con información sobre todos los comentarios asociados a ese objeto
Escenario alternativo	<ol style="list-style-type: none"> 1c. Se realiza una petición al servidor de la aplicación, especificando el nombre de usuario al cual pertenecen los comentarios 2c. Se recibe un objeto JSON con información sobre todos los comentarios asociados a ese nombre de usuario

Tabla 24. Caso de uso CU-09

3.3 Requisitos de software

Mediante los requisitos de software se detalla con un nivel de abstracción mas bajo la funcionalidad que se espera de la aplicación, así como sus limitaciones. Se dividen en dos clases:

- Requisitos funcionales: requisitos que definen la funcionalidad que la aplicación debe proporcionar.
- Requisitos no funcionales: requisito que especifican limitaciones sobre la funcionalidad anterior.

Para la especificación de los diferentes requisitos se utilizan los siguientes campos:

- **Identificador:** permite reconocer cada requisito de forma unívoca. Este campo cumple con la siguiente nomenclatura: RSF-XX en el caso de los requisitos funcionales y RSNF-XX en el caso de los requisitos no funcionales, donde XX se corresponde con dígitos numéricos. Este identificador no tiene la obligación de guardar relación con el identificador del requisito anterior.
- **Nombre:** describe de manera breve el objetivo del requisito.
- **Descripción:** especifica de manera completa en qué consiste el requisito.
- **Fuente:** especifica el origen del requisito.
- **Necesidad:** especifica el grado de importancia que tiene el requisito para el proyecto. Los valores que puede tomar este campo son: esencial, deseable u opcional.
- **Prioridad:** especifica la urgencia con la que el requisito debe cumplirse. Los valores que puede tomar este campo son: alta, media o baja.
- **Claridad:** especifica si el requisito es entendible por sí solo y de manera correcta. Los valores que puede tomar este campo son: alta, media o baja.
- **Estabilidad:** especifica si el requisito puede variar a lo largo de la vida del proyecto. Los valores que puede tomar este campo son: estable o inestable.
- **Verificabilidad:** especifica si el cumplimiento del requisito puede comprobarse. Los valores que puede tomar este campo son: alta, media o baja.

3.3.1 Usuario final

3.3.1.1 Requisitos funcionales

RSF-01			
Nombre	Selección e identificación de objeto real		
Descripción	Selección e identificación del objeto real, mediante el uso del lector de códigos RFID conectado a la aplicación		
Fuente	RUC-01	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 25. Requisito funcional RSF-01

RSF-02			
Nombre	Mostrar información sobre el objeto real seleccionado		
Descripción	La aplicación mostrará en pantalla una introducción breve sobre el objeto real seleccionado y su nombre, una vez sea identificado por su código RFID asociado		
Fuente	RUC-01	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 26. Requisito funcional RSF-02

RSF-03			
Nombre	Mostrar vídeos asociados al objeto real seleccionado		
Descripción	La aplicación mostrará en pantalla una lista de vídeos relacionados con el objeto real seleccionado una vez sea identificado por su código RFID asociado		
Fuente	RUC-01	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 27. Requisito funcional RSF-03

RSF-04			
Nombre	Seleccionar vídeo sobre el objeto real seleccionado		
Descripción	Seleccionar un vídeo de la lista de vídeos asociada al objeto seleccionado		
Fuente	RUC-02	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 28. Requisito funcional RSF-04

RSF-05			
Nombre	Reproducir vídeo seleccionado		
Descripción	Reproducir el vídeo seleccionado, aumentando el número de visitas del mismo		
Fuente	RUC-02	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 29. Requisito funcional RSF-05

RSF-06			
Nombre	Valorar vídeo seleccionado		
Descripción	Valorar positiva o negativamente el vídeo seleccionado		
Fuente	RUC-02	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 30. Requisito funcional RSF-06

RSF-07			
Nombre	Acceder a comentarios existentes		
Descripción	Acceder a la lista de comentarios relacionados con el vídeo seleccionado, creados previamente por otros usuarios		
Fuente	RUC-03	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 31. Requisito funcional RSF-07

RSF-08			
Nombre	Crear nuevo comentario		
Descripción	Crear nuevo comentario relacionado con el vídeo seleccionado, pudiendo incluir texto y vídeo como contenido del mismo. Será necesario también proporcionar un nombre de usuario		
Fuente	RUC-03	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 32. Requisito funcional RSF-08

RSF-09			
Nombre	Responder a un comentario ya existente		
Descripción	Responder a un comentario ya existente mediante la creación de un nuevo comentario, pudiendo incluir texto y vídeo y siendo necesario proporcionar un nombre de usuario		
Fuente	RUC-03	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 33. Requisito funcional RSF-09

RSF-10			
Nombre	Crear nuevo vídeo		
Descripción	Creación de un nuevo vídeo asociado al objeto real seleccionado		
Fuente	RUC-04	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 34. Requisito funcional RSF-10

3.3.1.2 Requisitos no funcionales

Requisitos operacionales

RSNF-01			
Nombre	Formato de archivos de vídeo		
Descripción	El formato de los archivos de vídeo es WebM		
Fuente	RUC-02, RUC-04, RUR-03	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 35. Requisito no funcional RSNF-01

RSNF-02			
Nombre	Navegador		
Descripción	La aplicación será compatible con el navegador webG Chrome, debido al requisito de creación de vídeos mediante el uso de una cámara externa		
Fuente	RUC-02, RUC-04, RUR-02	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 36. Requisito no funcional RSNF-02

RSNF-03			
Nombre	Sistema operativo		
Descripción	La aplicación funcionará en el entorno operativo Windows y OS X		
Fuente	RUR-04	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 37. Requisito no funcional RSNF-03

Requisitos de interfaz

RSNF-04			
Nombre	Idioma		
Descripción	El idioma de la aplicación es el Inglés		
Fuente	RUR-01	Necesidad	Media
Prioridad	Media	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 38. Requisito no funcional RSNF-04

RSNF-05			
Nombre	Iconos y botones de la interfaz		
Descripción	Iconos y botones grandes e intuitivos		
Fuente	RUR-06	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 39. Requisito no funcional RSNF-05

RSNF-06			
Nombre	Utilización de la interfaz		
Descripción	Interfaz fácil de utilizar mediante el uso de botones e iconos descriptivos, fuente y controles de tamaño razonable y fácil navegación		
Fuente	RUR-06	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 40. Requisito no funcional RSNF-06**Requisitos de comprobación**

RSNF-07			
Nombre	Comprobar cumplimentación de los campos		
Descripción	Comprobar que se han rellenado todos los campos necesarios a la hora de crear un nuevo vídeo o comentario		
Fuente	RUC-03, RUC-04	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 41. Requisito no funcional RSNF-07

3.3.2 Desarrollador

3.3.2.1 Requisitos funcionales

RSF-11			
Nombre	Recolección de información de todos los objetos reales		
Descripción	Recolectar información sobre todos los objetos reales a los que da soporte la aplicación, mediante el uso de la API desarrollada		
Fuente	RUC-05	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 42. Requisito funcional RSF-11

RSF-12			
Nombre	Recolección de información de los objetos reales según su RFID		
Descripción	Recolectar información sobre los objetos reales cuyo RFID coincida con el utilizado en la búsqueda		
Fuente	RUC-05	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 43. Requisito funcional RSF-12

RSF-13			
Nombre	Recolección de información de los objetos reales según su nombre		
Descripción	Recolectar información sobre los objetos reales cuyo nombre coincida con el utilizado en la búsqueda		
Fuente	RUC-05	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 44. Requisito funcional RSF-13

RSF-14			
Nombre	Recolección de información de todos los vídeos almacenados		
Descripción	Recolectar todos los vídeos creados por usuarios almacenados en la aplicación		
Fuente	RUC-06	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 45. Requisito funcional RSF-14

RSF-15			
Nombre	Recolección de vídeos por RFID		
Descripción	Recolectar todos los vídeos asociados a un objeto real, mediante la especificación de su RFID en la búsqueda		
Fuente	RUC-06	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 46. Requisito funcional RSF-15

RSF-16			
Nombre	Recolección de vídeos por nombre		
Descripción	Recolectar todos los vídeos asociados a un objeto real, mediante la especificación de su nombre en la búsqueda		
Fuente	RUC-06	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 47. Requisito funcional RSF-16

RSF-17			
Nombre	Recolección de todos los comentarios almacenados		
Descripción	Recolectar todos los comentarios creados por usuarios almacenados en la aplicación		
Fuente	RUC-07	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 48. Requisito funcional RSF-17

RSF-18			
Nombre	Recolección de comentarios por RFID		
Descripción	Recolectar todos los comentarios asociados a un objeto mediante la especificación de su RFID en la búsqueda		
Fuente	RUC-07	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 49. Requisito funcional RSF-18

RSF-19			
Nombre	Recolección de comentarios por nombre de objeto		
Descripción	Recolectar todos los comentarios asociados a un objeto mediante la especificación de su nombre en la búsqueda		
Fuente	RUC-07	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 50. Requisito funcional RSF-19

RSF-20			
Nombre	Recolección de comentarios por nombre de usuario		
Descripción	Recolectar todos los comentarios existentes asociados a un usuario, mediante la especificación del nombre de dicho usuario en la búsqueda		
Fuente	RUC-07	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 51. Requisito funcional RSF-20

3.3.2.2 Requisitos no funcionales

Requisitos operacionales

RSNF-08			
Nombre	Formato de información recolectada		
Descripción	El formato de la información recolectada del servidor es JSON (JavaScript Object Notation)		
Fuente	RUC-05, RUC-06, RUC-07, RUR-07	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 52. Requisito no funcional RSNF-08

Requisitos de comprobación

RSNF-09			
Nombre	Comprobación de información recolectada		
Descripción	Comprobar en la aplicación, una vez se esté recolectando la información requerida, que dicha información existe y no contiene errores		
Fuente	RUC-05, RUC-06, RUC-07	Necesidad	Alta
Prioridad	Alta	Claridad	Alta
Estabilidad	Estable	Verificabilidad	Alta

Tabla 53. Requisito no funcional RSNF-09

3.4 Matrices de trazabilidad

Mediante el empleo de matrices de trazabilidad se puede comprobar de manera más sencilla que todos los requisitos de usuario han sido cubiertos tanto en la especificación de uso como de requisitos de software.

3.4.1 Usuario final

3.4.1.1 Matriz de trazabilidad entre requisitos de usuario y casos de uso

REQUISITOS DE USUARIO	CASOS DE USO						
REQUISITOS DE CAPACIDAD	CU-01	CU-02	CU-03	CU-04	CU-05	CU-06	CU-07
RUC-01	X						
RUC-02		X	X		X		
RUC-03						X	X
RUC-04				X			
REQUISITOS DE RESTRICCIÓN	CU-01	CU-02	CU-03	CU-04	CU-05	CU-06	CU-07
RUR-01	X	X	X	X	X	X	X
RUR-02	X	X	X	X	X	X	X
RUR-03		X	X	X	X		
RUR-04	X	X	X	X	X	X	X
RUR-05	X						
RUR-06	X	X	X	X	X	X	X

Tabla 54. Matriz de trazabilidad entre requisitos de usuario y casos de uso (usuario final)

3.4.1.2 Matriz de trazabilidad entre requisitos de usuario y requisitos de software funcionales

REQUISITOS DE USUARIO	REQUISITOS DE SOFTWARE FUNCIONALES									
REQUISITOS DE CAPACIDAD	RSF-10	RSF-02	RSF-03	RSF-04	RSF-05	RSF-06	RSF-07	RSF-08	RSF-09	RSF-10
RUC-01	X	X								
RUC-02			X	X	X	X				
RUC-03							X	X	X	
RUC-04										X
REQUISITOS DE RESTRICCIÓN	RSF-10	RSF-02	RSF-03	RSF-04	RSF-05	RSF-06	RSF-07	RSF-08	RSF-09	RSF-10
RUR-01	X	X	X	X	X	X	X	X	X	X
RUR-02	X	X	X	X	X	X	X	X	X	X
RUR-03			X	X	X	X				
RUR-04	X	X	X	X	X	X	X	X	X	X
RUR-05	X									
RUR-06	X	X	X	X	X	X	X	X	X	X

Tabla 55. Matriz de trazabilidad entre requisitos de usuario requisitos de software funcionales (usuario final)

3.4.1.3 Matriz de trazabilidad entre requisitos de usuario y requisitos de software funcionales

REQUISITOS DE USUARIO	REQUISITOS DE SOFTWARE NO FUNCIONALES						
REQUISITOS DE CAPACIDAD	RSNF-01	RSNF-02	RSNF -03	RSNF -04	RSNF -05	RSNF -06	RSNF -07
RUC-01		X	X	X	X	X	
RUC-02	X	X	X	X	X	X	
RUC-03		X	X	X	X	X	X
RUC-04	X	X	X	X	X	X	X
REQUISITOS DE RESTRICCIÓN	RSNF -01	RSNF -02	RSNF -03	RSNF -04	RSNF -05	RSNF -06	RSNF -07
RUR-01		X	X	X	X	X	
RUR-02		X	X	X	X	X	
RUR-03	X	X	X	X	X	X	
RUR-04		X	X	X	X	X	
RUR-05		X	X	X	X	X	
RUR-06		X	X	X	X	X	

Tabla 56. Matriz de trazabilidad entre requisitos de usuario requisitos de software no funcionales (usuario final)

3.4.2 Desarrollador

3.4.2.1 Matriz de trazabilidad entre requisitos de usuario y casos de uso

REQUISITOS DE USUARIO	CASOS DE USO		
REQUISITOS DE CAPACIDAD	CU-08	CU-09	CU-10
RUC-05	X		
RUC-06		X	
RUC-07			X
REQUISITOS DE RESTRICCIÓN	CU-08	CU-09	CU-10
RUR-07	X	X	X

Tabla 57. Matriz de trazabilidad entre requisitos de usuario y casos de uso (desarrollador)

3.4.2.2 Matriz de trazabilidad entre requisitos de usuario y requisitos de software funcionales

REQUISITOS DE USUARIO	REQUISITOS DE SOFTWARE FUNCIONALES								
REQUISITOS DE CAPACIDAD	RSF-11	RSF-12	RSF-13	RSF-14	RSF-15	RSF-16	RSF-17	RSF-18	RSF-19
RUC-05	X	X	X						
RUC-06				X	X	X			
RUC-07							X	X	X
REQUISITOS DE RESTRICCIÓN	RSF-11	RSF-12	RSF-13	RSF-14	RSF-15	RSF-16	RSF-17	RSF-18	RSF-19
RUR-07	X	X	X	X	X	X	X	X	X

Tabla 58. Matriz de trazabilidad entre requisitos de usuario y requisitos de software funcionales (desarrollador)

3.4.2.3 Matriz de trazabilidad entre requisitos de usuario y requisitos de software funcionales

REQUISITOS DE USUARIO	REQUISITOS DE SOFTWARE NO FUNCIONALES	
REQUISITOS DE CAPACIDAD	RSNF-08	RSNF-09
RUC-05	X	X
RUC-06	X	X
RUC-07	X	X
REQUISITOS DE RESTRICCIÓN	RSNF-08	RSNF-09
RUR-07	X	X

Tabla 59. Matriz de trazabilidad entre requisitos de usuario y requisitos de software no funcionales (desarrollador)

Capítulo 4

Diseño

Posteriormente a la fase de análisis se procede al diseño del sistema. Esta fase comienza especificando el contexto en el que se integra el sistema a diseñar. Posteriormente se incluye una justificación de la arquitectura de software elegida, continuando con la especificación de los componentes de software necesarios para el cumplimiento de la funcionalidad. Tras este paso, se realiza la matriz de trazabilidad con el fin de asegurar que toda la funcionalidad se encuentra incluida en dichos componentes. Para finalizar, se incluye el diseño de la API para desarrolladores y un conjunto de prototipos de bajo nivel de la interfaz de la aplicación.

4.1 Contexto del sistema

Para comenzar el diseño de la aplicación, es necesario obtener una visión general del sistema y conocer el marco en el que se encuentra, los sistemas con los que interactúa y sus límites.

Tanto en el contexto de un usuario final como en el de un desarrollador, la aplicación depende siempre de la información almacenada en la base de datos. De este modo, los objetos permitidos, vídeos, comentarios, valoraciones y visualizaciones están presentes en dicha base de datos, siendo esta el principal componente que permite el funcionamiento de la aplicación.

La estructura de la base de datos de la aplicación será descrita más adelante, siendo un requisito esencial para un almacenamiento correcto de los datos generados por la aplicación.

Es importante señalar que, en el caso del uso por parte de un usuario final, la comunicación entre la aplicación y la base de datos se da de forma continua, de forma que los datos generados por la aplicación se almacenan en tiempo real.

En el caso del uso por parte de un desarrollador, la comunicación con la base de datos no es continua, sino puntual, ya que este tipo de usuario no genera datos para almacenar, sino que realiza acciones de recolección de datos almacenados en la base de datos de la aplicación.

En la siguiente ilustración se puede apreciar de forma clara el contexto de la aplicación:

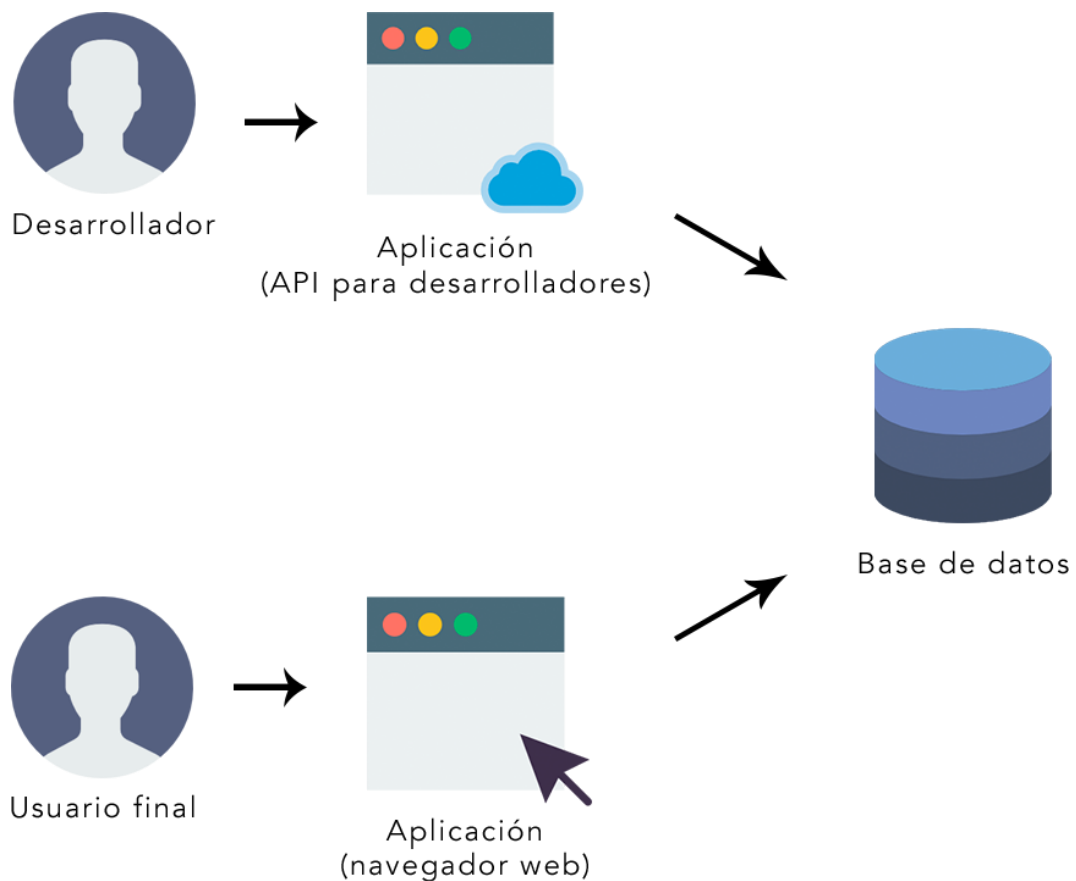


Ilustración 6. Contexto de la aplicación

4.2 Arquitectura de software

Mediante la definición de la arquitectura del software se permite tener una visión general de la estructura del sistema conociendo sus componentes y las comunicaciones entre ellos.

En este caso, existen tres partes claramente diferenciadas: las interfaces que permiten las interacciones por parte de los usuarios, los datos generados por el uso de la aplicación, y la lógica que permite la unión de las partes anteriores. De esto, se puede deducir que la arquitectura apropiada para la aplicación es la denominada Modelo Vista Controlador (MVC).

La arquitectura Modelo Vista Controlador consta de tres capas, como puede apreciarse en la ilustración 7, cada una con una función diferenciada, que se detallan a continuación:

- Modelo: representa los datos con los que trabaja la aplicación y permite la independencia de los mismos con respecto a su representación.
- Vista: representa la interfaz que permite al usuario interactuar con la aplicación, obteniendo una representación de los datos.
- Controlador: actúa como intermediario entre las capas anteriores, recibiendo peticiones de la vista, generadas mediante la interacción del usuario, que implican cambios en el modelo. Estos cambios son notificados a su vez a la vista para su actualización.

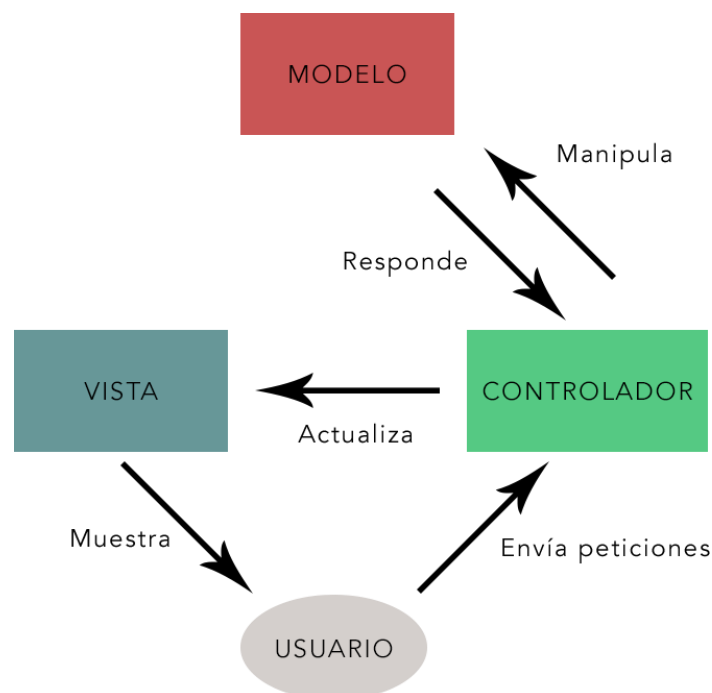


Ilustración 7. Arquitectura Modelo Vista Controlador

Esta división en capas permite separar la lógica de la aplicación de la interfaz, proporcionando modularidad y, por consiguiente, minimizando las dependencias entre las partes. De este modo, se facilita su mantenimiento y se aumenta la capacidad de reutilización de las partes, así como su flexibilidad.

4.2.1 Arquitectura del sistema

La funcionalidad del sistema se encuentra dividida en diversos componentes ubicados en cada una de las capas de la arquitectura. Para obtener estos componentes, se tienen en cuenta los requisitos obtenidos, dividiéndolos en grupos según su funcionalidad, y asignando estos grupos a componentes que se encarguen de la funcionalidad. Además, es necesario tener presente las relaciones que existen entre los elementos principales del sistema y, por tanto, las comunicaciones entre los mismos.

4.2.1.1 Usuario final

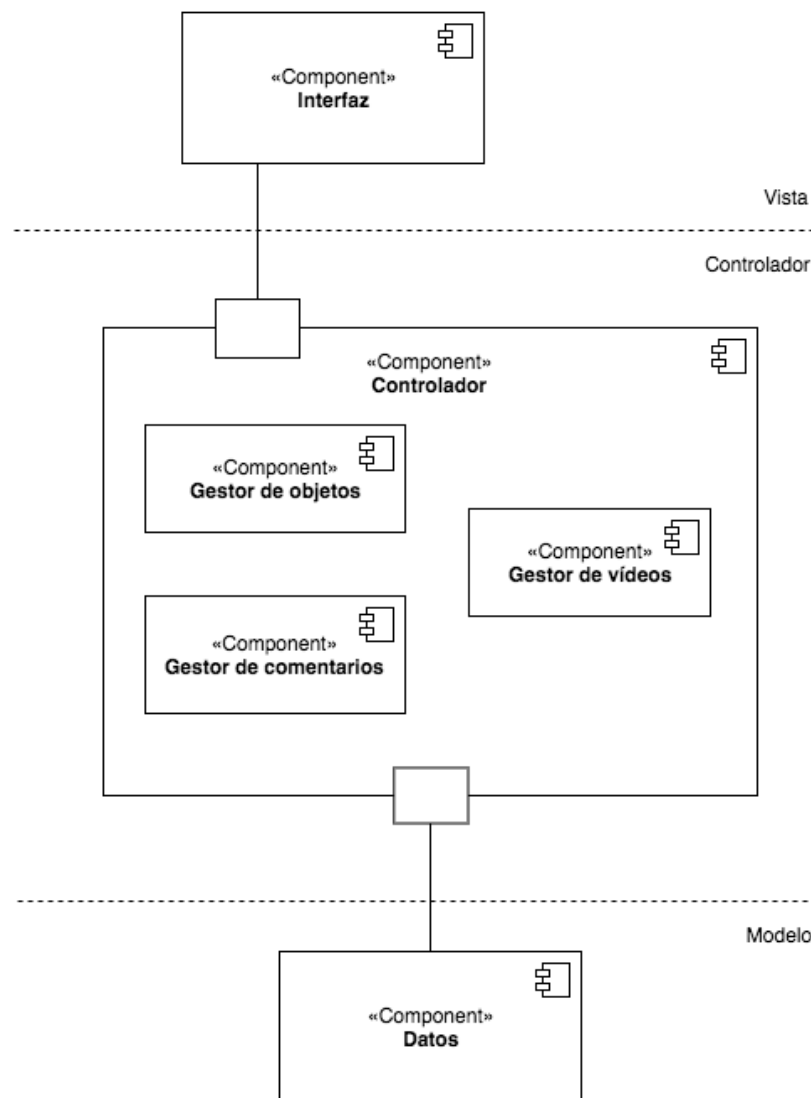


Ilustración 8. Diagrama de componentes (usuario final)

Vista

En la capa correspondiente a la vista se ubica un único componente denominado Interfaz. Este componente se encarga de presentar toda la información al usuario, utilizando distintas interfaces y proporcionarle un medio de interacción, actuando de intermediario entre dicho usuario y el controlador.

Cuando el usuario final interactúa con la interfaz, la información que proporciona (nuevos vídeos, comentarios, reproducciones y valoraciones) se comunica al controlador para que este pueda notificar al modelo y se realicen los cambios pertinentes.

Controlador

El controlador del sistema se divide en un grupo de componentes, cada uno de los cuales se ocupa de una funcionalidad diferente. Cada componente se comunica con la vista de manera independiente, proporcionando una interfaz específica mediante la cual recibe los datos que genera el usuario y realiza las acciones pertinentes sobre el modelo.

Estos componentes que conforman la capa del controlador son:

- Gestor de objetos: este componente se encarga de las acciones relativas a la recolección de datos sobre objetos reales elegidos por los usuarios para iniciar la aplicación.
- Gestor de vídeos: este componente se encarga cualquier acción relativa a los vídeos: valoraciones, reproducciones, creación de nuevos vídeos y recolección de vídeos asociados al objeto real seleccionado.
- Gestor de comentarios: este componente está a cargo de administrar las acciones posibles con comentarios, ya sea la creación de uno nuevo, recolectar los comentarios asociados a un vídeo o las respuestas a comentarios ya existentes.

Modelo

En la capa de modelo de la aplicación se ubica un único componente, denominado Datos. Como su nombre indica, es el encargado de almacenar los datos generados por la aplicación, garantizando su persistencia y disponibilidad.

En el caso de la información relacionada con los objetos reales a los que la aplicación da soporte, esta deberá ser introducida manualmente, ya que no es un tipo de información que pueda ser generada por los usuarios finales y dependerá directamente del evento en cuestión.

4.2.1.2 Desarrollador

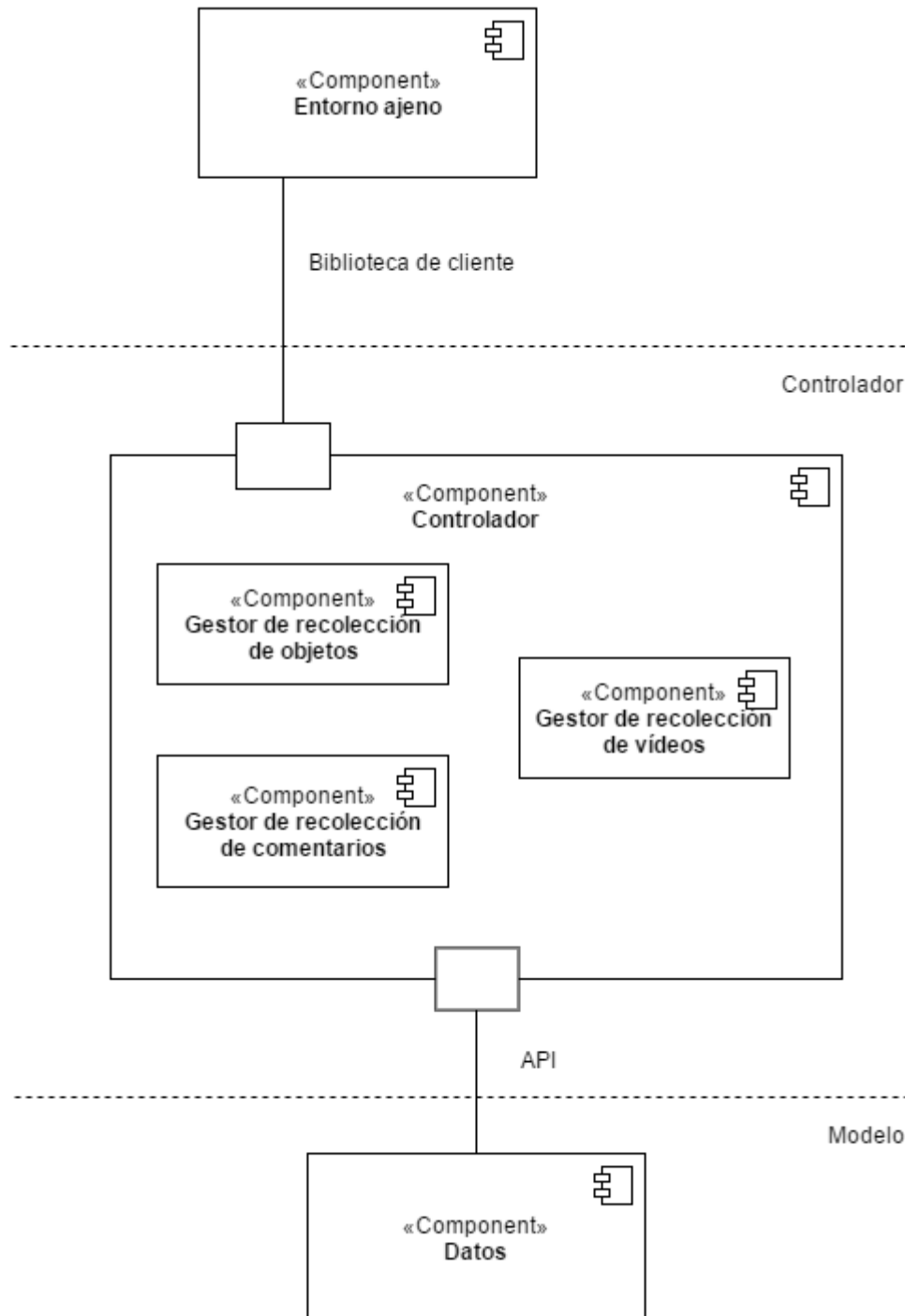


Ilustración 9. Diagrama de componentes (desarrollador)

En el caso del uso de la aplicación por parte de un desarrollador, los componentes involucrados varían con respecto a los de un usuario final.

Entorno ajeno

Este componente hace referencia al entorno donde el desarrollador quiera plasmar los datos recolectados, pudiendo ser el caso de una aplicación que tome datos de esta misma, o una investigación relacionada con este proyecto. Este componente se comunicará, mediante una biblioteca de cliente, con el controlador encargado de proporcionar la API de recolección de datos, mediante la cual el desarrollador en cuestión podrá realizar peticiones de búsqueda en la base de datos de la aplicación.

Es importante señalar que las acciones llevadas a cabo por parte del desarrollador no implicarán ningún cambio en los datos almacenados, debido a que las únicas acciones posibles son de recolección.

Controlador

El controlador encargado de proporcionar al desarrollador la funcionalidad de recolección de datos de la aplicación difiere del controlador del cliente principalmente en las acciones que permite llevar a cabo sobre la aplicación. En este caso, el desarrollador no podrá hacer uso de la interfaz de la aplicación web, al igual que tampoco podrá crear nuevos datos para su posterior almacenamiento en la base de datos de la aplicación. Por lo tanto, la funcionalidad que este controlador brindará al desarrollador es únicamente la de recolección de datos, mediante el uso de la API diseñada e implementada para este fin.

Modelo

La capa de modelo es compartida tanto en los casos de uso de un usuario final como en los de un desarrollador.

4.3 Especificación del diseño de componentes

Para sintetizar y completar la información proporcionada en el apartado anterior y facilitar la comprobación de la trazabilidad con los requisitos capturados en la fase de

análisis, se proporciona una especificación de los diferentes componentes en forma de tablas en las que se utilizan los siguientes campos:

- Identificador: permite reconocer cada componente de manera unívoca. Este campo cumple con la siguiente nomenclatura: CO-XX, donde XX se corresponde con dígitos numéricos.
- Nombre: especifica el nombre del componente para que pueda ser identificado en la ilustración correspondiente a la arquitectura del sistema.
- Tipo: especifica el nivel al que se encuentra el componente
- Propósito: especifica los requisitos de software cuya funcionalidad es implementada en el componente.
- Función: especifica las funcionalidades que lleva a cabo el componente
- Subordinados: especifica los componentes en los que se divide el componente actual

4.3.1 Usuario final

CO-01	
Nombre	Interfaz
Tipo	Subsistema
Propósito	RSF-02, RSF-03, RSF-04, RSF-05, RSF-06, RSF-07, RSF-08, RSF-09, RSF-10
Función	Permite la interacción entre el usuario y la aplicación, transmitiendo las peticiones al controlador y recibiendo los cambios
Subordinados	Ninguno

Tabla 60. Componente CO-01

CO-02	
Nombre	Controlador
Tipo	Subsistema
Propósito	RSF-01, RSF-02, RSF-03, RSF-04, RSF-05, RSF-06, RSF-07, RSF-08, RSF-09, RSF-10
Función	Recibe y procesa las peticiones de la vista, invocando al modelo, el cual puede sufrir cambios, que se notificarán de nuevo a la vista
Subordinados	<ul style="list-style-type: none"> - Gestor de objetos - Gestor de vídeos - Gestor de comentarios

Tabla 61. Componente CO-02

CO-03	
Nombre	Gestor de objetos
Tipo	Componente
Propósito	RSF-01, RSF-02, RSF-10
Función	Permite toda funcionalidad relacionada con los objetos reales y sus datos correspondientes en la base de datos
Subordinados	Ninguno

Tabla 62. Componente CO-03

CO-04	
Nombre	Gestor de vídeos
Tipo	Componente
Propósito	RSF-03, RSF-04, RSF-05, RSF-06, RSF-10
Función	Permite toda funcionalidad relacionada con los vídeos correspondientes a objetos reales y a la creación de nuevos vídeos
Subordinados	Ninguno

Tabla 63. Componente CO-04

CO-05	
Nombre	Gestor de comentarios
Tipo	Componente
Propósito	RSF-07, RSF-08, RSF-09
Función	Permite toda funcionalidad relacionada con los comentarios correspondientes a vídeos existentes y a la creación de nuevos comentarios
Subordinados	Ninguno

Tabla 64. Componente CO-05

CO-06	
Nombre	Datos
Tipo	Subsistema
Propósito	RSF-01, RSF-02, RSF-03, RSF-04, RSF-05, RSF-06, RSF-07, RSF-08, RSF-09, RSF-10
Función	Permite toda la funcionalidad relacionada con el almacenamiento y recuperación de información de la aplicación
Subordinados	Ninguno

Tabla 65. Componente CO-06

4.3.2 Desarrollador

CO-07	
Nombre	Controlador
Tipo	Subsistema
Propósito	RSF-11, RSF-12, RSF-13, RSF-14, RSF-15, RSF-16, RSF-17, RSF-18, RSF-19, RSF-20
Función	Recibe las peticiones por parte del desarrollador, y lleva a cabo las consultas necesarias en la base de datos de la aplicación
Subordinados	<ul style="list-style-type: none"> - Gestor de recolección de objetos - Gestor de recolección de vídeos - Gestor de recolección de comentarios

Tabla 66. Componente CO-07

CO-08	
Nombre	Gestor de recolección de objetos
Tipo	Componente
Propósito	RSF-11, RSF-12, RSF-13
Función	Proporciona toda la funcionalidad relacionada a la recolección de información sobre objetos reales almacenada en la aplicación
Subordinados	Ninguno

Tabla 67. Componente CO-08

CO-09	
Nombre	Gestor de recolección de vídeos
Tipo	Componente
Propósito	RSF-14, RSF-15, RSF-16
Función	Proporciona toda la funcionalidad relacionada a la recolección de información sobre vídeos almacenada en la aplicación
Subordinados	Ninguno

Tabla 68. Componente CO-08

CO-10	
Nombre	Gestor de recolección de comentarios
Tipo	Componente
Propósito	RSF-17, RSF-18, RSF-19, RSF-20
Función	Proporciona toda la funcionalidad relacionada a la recolección de información sobre comentarios almacenada en la aplicación
Subordinados	Ninguno

Tabla 69. Componente CO-09

4.4 Matrices de trazabilidad

Mediante el empleo de matrices de trazabilidad puede comprobarse de manera más sencilla que todos los requisitos de software han sido cubiertos en la especificación de componentes.

4.4.1 Usuario final

Requisitos de software funcionales	Componentes					
	CO-01	CO-02	CO-03	CO-04	CO-05	CO-06
RSF-01		X	X			X
RSF-02	X	X	X			X
RSF-03	X	X		X		X
RSF-04	X	X		X		X
RSF-05	X	X		X		X
RSF-06	X	X		X		X
RSF-07	X	X			X	X
RSF-08	X	X			X	X
RSF-09	X	X			X	X
RSF-10	X	X	X	X		X

Tabla 70. Tabla de trazabilidad entre requisitos de software y componentes (usuario final)

4.4.2 Desarrollador

Requisitos de software funcionales	Componentes			
	CO-07	CO-08	CO-09	CO-10
RSF-11	X	X		
RSF-12	X	X		
RSF-13	X	X		
RSF-14	X		X	
RSF-15	X		X	
RSF-16	X		X	
RSF-17	X			X
RSF-18	X			X
RSF-19	X			X
RSF-20	X			X

Tabla 71. Tabla de trazabilidad entre requisitos de software y componentes (desarrollador)

4.5 Prototipos de la interfaz

Mediante el uso de prototipos es posible obtener una versión preliminar de la apariencia de la interfaz de usuario, así como de la funcionalidad incluida en la aplicación. Gracias a esta técnica, el cliente tiene una toma de contacto con la futura aplicación previamente a su desarrollo final, lo que permite su evaluación y, por tanto, reduce o elimina las disconformidades y cambios en fases futuras. A continuación, se muestran los prototipos de bajo nivel realizados para las distintas pantallas de la aplicación.

4.5.1 Pantalla principal



Ilustración 10. Prototipo de bajo nivel para la pantalla principal

Cuando la aplicación esté ejecutándose, se muestra en pantalla el nombre de la aplicación, a la espera de que el usuario seleccione un objeto real y lo coloque sobre el lector de códigos RFID, permitiendo a la aplicación identificar dicho objeto y mostrar información sobre el mismo.

4.5.2 Pantalla de objeto seleccionado

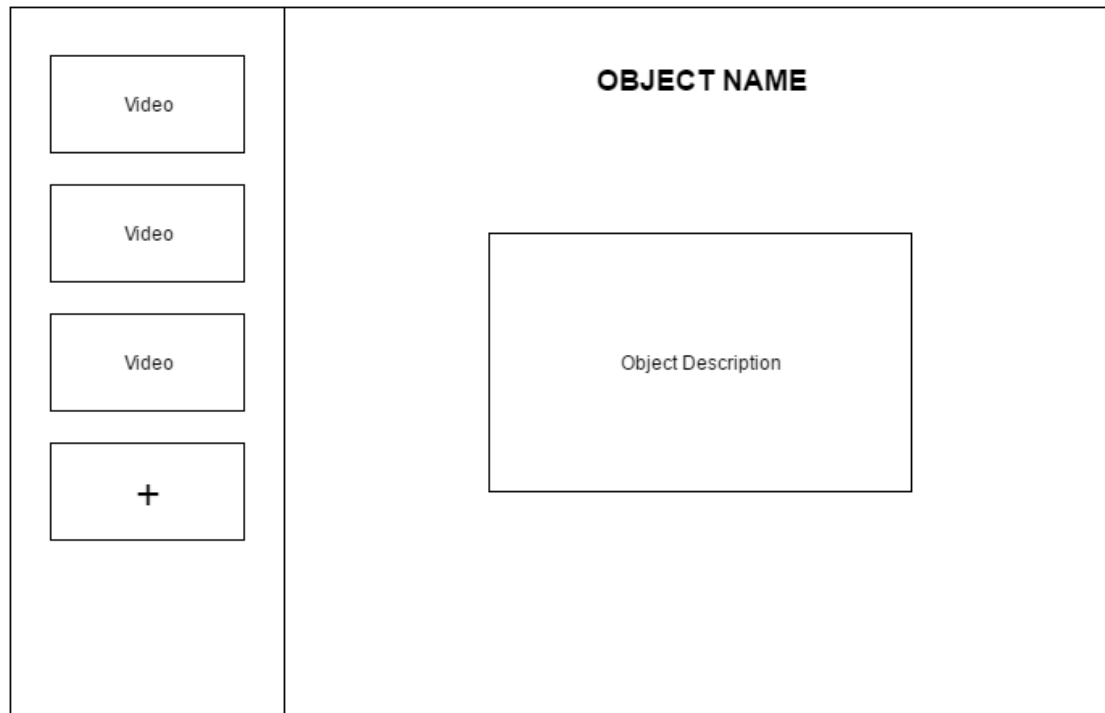


Ilustración 11. Prototipo de bajo nivel para la pantalla de objeto seleccionado

Tras seleccionar el objeto real y habiéndolo posicionado sobre el lector de códigos RFID, la aplicación identificará dicho objeto y mostrará información relacionado con el mismo. Como se puede observar en la ilustración 10, dicha información consiste en una lista de vídeos asociados al objeto, posicionada a la izquierda de la pantalla, e información básica del propio objeto, como puede ser su nombre y su descripción, posicionada en la zona central de la pantalla.

La sección donde se encuentra la lista de vídeos en la pantalla será el punto de partida para interactuar con la aplicación, ya que tendrá el papel de menú principal, desde el cual se podrá seleccionar cualquiera de los vídeos asociados al objeto seleccionado (pudiendo llevar a cabo más operaciones tras haber hecho esto) y también elegir la opción de crear y enviar un nuevo vídeo.

4.5.3 Pantalla de vídeo seleccionado

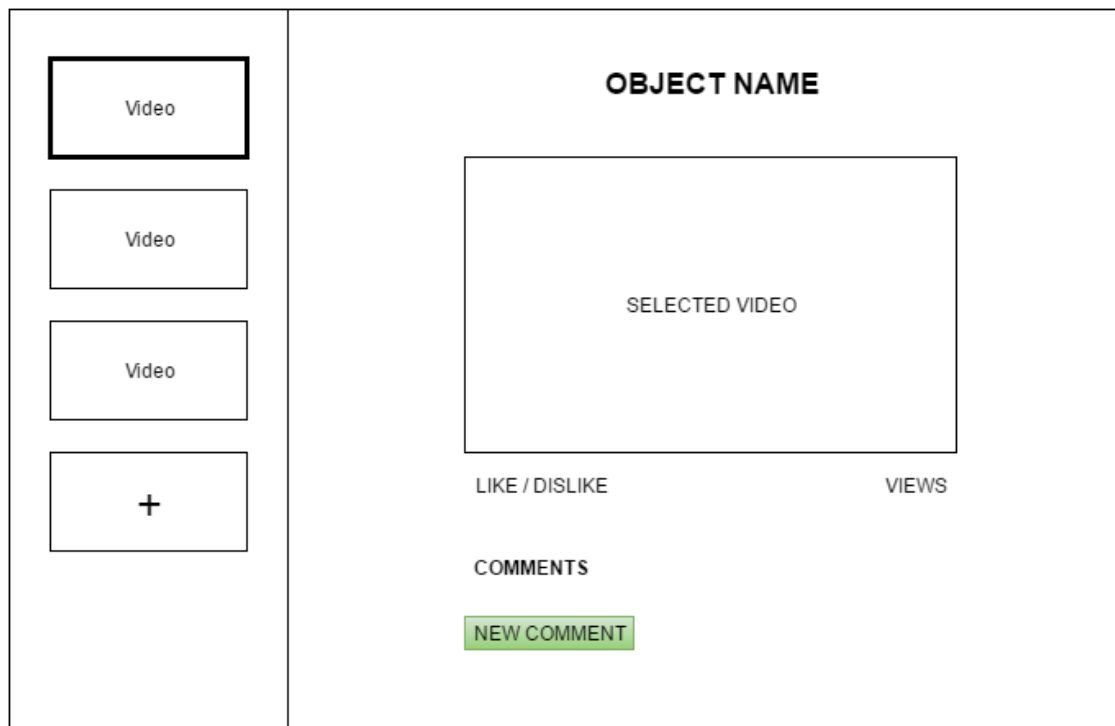


Ilustración 12. Prototipo de bajo nivel para la pantalla de vídeo seleccionado

Tras haber sido seleccionado el objeto real para comenzar a usar la aplicación, una de las principales acciones para llevar a cabo por parte del usuario es la de selección de un vídeo asociado a dicho objeto real, pulsando sobre cualquiera de los vídeos listados en la franja izquierda de la pantalla.

Una vez se haya seleccionado el vídeo deseado, la zona central de la pantalla cambiará su contenido, mostrando:

- El vídeo seleccionado, el cual podrá reproducirse pulsando sobre el botón del reproductor
- Los botones de valoración del vídeo, pudiendo interactuar con ellos con el fin de valorar de forma permanente el vídeo seleccionado
- El número de visitas del vídeo, el cual aumentará por cada reproducción que reciba el mismo
- La sección de comentarios, la cual será explicada a continuación

4.5.4 Pantalla de vídeo seleccionado, sección de comentarios

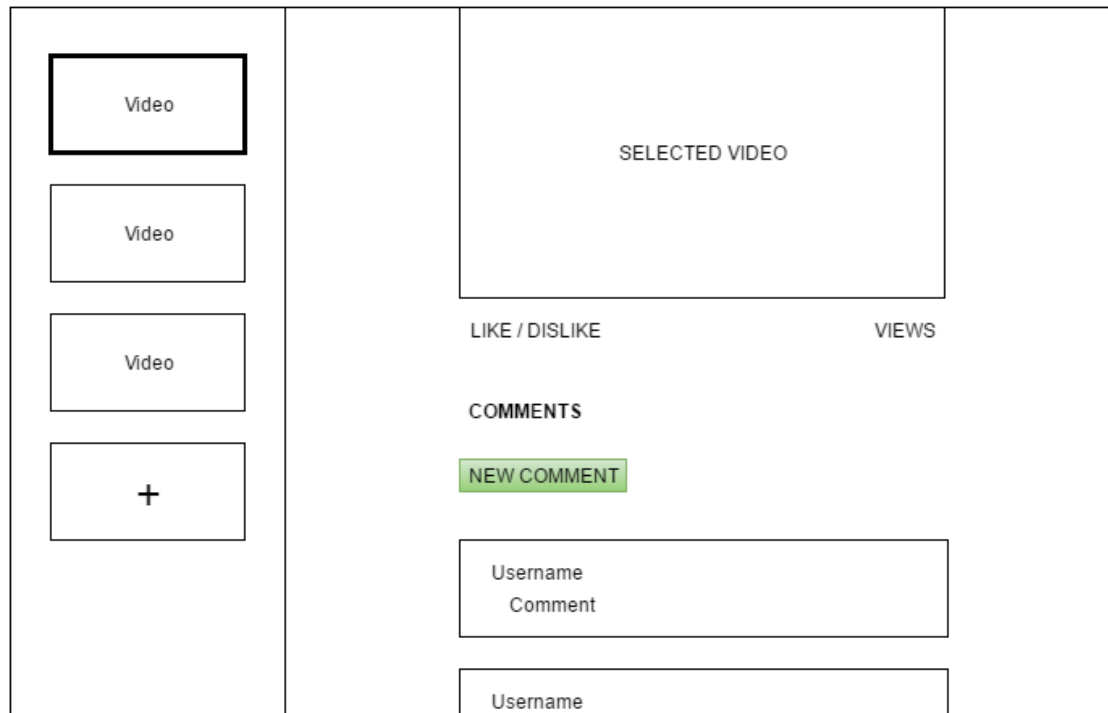


Ilustración 13. Prototipo de bajo nivel para la sección de comentarios, en la pantalla de vídeo seleccionado

En la misma pantalla de vídeo seleccionado se encuentra también la sección de comentarios relacionados con dicho vídeo, ya existentes en la base de datos de la aplicación. A su vez, en esta sección se encuentra el botón diseñado para permitir al usuario crear un nuevo comentario, creando así nuevo contenido para ser almacenado en la base de datos.

Además de poder revisar los comentarios existentes y crear nuevos, el usuario también podrá responder a cualquier comentario listado, mediante el uso del botón de respuesta asociado a cada comentario. Esta acción es muy similar a crear un nuevo comentario asociado al vídeo, pero siendo almacenado no como un comentario aislado, sino como una respuesta incluida dentro del comentario original, al cual se referencia.

4.5.5 Pantalla de creación de nuevo comentario

NEW COMMENT

Username

Comment

SUBMIT

RECORD STOP

Video

Ilustración 14. Prototipo de bajo nivel para la pantalla de creación de nuevo comentario

Siguiendo el orden de interacción por parte del usuario final, el siguiente paso tras seleccionar uno de los vídeos asociados al objeto real es la creación de contenido basado en dicho vídeo. Tras haber pulsado sobre el botón de creación de nuevo comentario o sobre el botón de respuesta a un comentario ya existente, la aplicación sufrirá un cambio en la interfaz, mostrando una sección nueva, ocupando la totalidad de la pantalla.

Esta sección permitirá al usuario crear un nuevo comentario, mediante un formulario con dos datos: el nombre del usuario y el contenido del comentario. A su vez, el usuario también podrá grabar un vídeo para añadirlo al contenido del comentario. Es importante recalcar que este vídeo no formará parte de los vídeos asociados al objeto real seleccionado, sino que estará únicamente vinculado al nuevo comentario.

Tras haber terminado de añadir la información deseada al contenido del comentario, el usuario podrá enviar el nuevo contenido generado a la aplicación, la cual se encargará de almacenarlo. Tras haber sido almacenado con éxito, la pantalla mostrará un mensaje, notificando al usuario que el comentario ha sido almacenado, y seguidamente la sección de creación de nuevo comentario se desvanecerá, volviendo a la pantalla de selección de vídeo, donde el nuevo comentario creado estará presente en la lista de comentarios asociados al vídeo.

Cabe destacar que el usuario podrá cancelar la creación de un nuevo comentario cerrando la sección de creación de nuevo comentario, pulsando sobre el botón diseñado para tal fin, localizado en la esquina superior derecha de la pantalla.

4.5.6 Pantalla de creación de nuevo vídeo

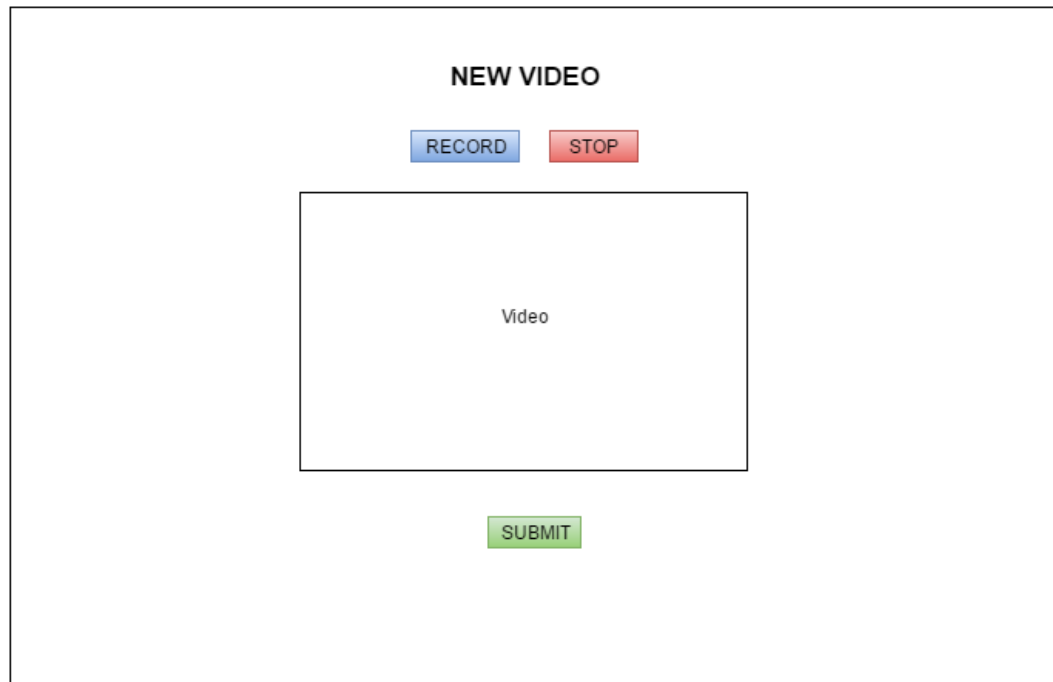


Ilustración 15. Prototipo de bajo nivel para la pantalla de creación de nuevo vídeo

Para finalizar, una de las acciones más importantes para el crecimiento de la aplicación que el usuario final puede llevar a cabo es la creación de nuevos vídeos asociados al objeto real seleccionado. Esto puede llevarse a cabo pulsando sobre el botón de creación de nuevo vídeo, localizado en el menú principal de la aplicación, bajo la lista de vídeos existentes.

Tras pulsar sobre dicho botón, la aplicación mostrará una nueva sección, similar a la anteriormente descrita para la creación de nuevos comentarios. En esta nueva sección, el usuario podrá grabar un nuevo vídeo pulsando en el botón de grabación, y podrá finalizar la grabación pulsando sobre el botón de finalización de vídeo.

Una vez el vídeo esté grabado, el usuario podrá visualizarlo, con el fin comprobar su correcta grabación. Seguidamente a esto, pulsando sobre el botón de enviar, el usuario enviará a la aplicación el nuevo vídeo grabado, almacenándose de esta forma en la base de datos. Tras esto, un mensaje de éxito se mostrará en pantalla.

Al igual que en la sección de creación de nuevos comentarios, el usuario podrá regresar a la pantalla anterior, mediante el uso del botón diseñado para ello, presente en la esquina superior derecha de la pantalla.

4.6 Diseño de API pública

Con el fin de proporcionar a desarrolladores e investigadores la posibilidad de recolectar la información almacenada en la base de datos de la aplicación, es necesario diseñar e implementar una API pública comprensible y descriptiva.

A continuación, se muestra el diseño final de la API pública, especificando las posibles operaciones de recolección de datos.

4.6.1 Objetos

Método	HTTP Request	Descripción
Get all	GET /objects/all	Recolecta información de todos los objetos existentes
Get by name	GET /objects/byName/:name	Recolecta información de todos los objetos asociados al nombre proporcionado

Tabla 72. Métodos de recolección de datos sobre objetos reales

4.6.2 Vídeos

Método	HTTP Request	Descripción
Get all	GET /videos/all	Recolecta información de todos los vídeos existentes en la base de datos
Get by object name	GET /videos/byObjectName/:name	Recolecta información de todos los vídeos asociados al nombre del objeto real proporcionado

Tabla 73. Métodos de recolección de datos sobre vídeos

4.6.3 Comentarios

Método	HTTP Request	Descripción
Get all	GET /comments/all	Recolecta información de todos los comentarios existentes en la base de datos
Get by username	GET /comments/byUsername/:name	Recolecta información de todos los comentarios asociados al nombre de usuario proporcionado

Tabla 74. Métodos de recolección de datos sobre comentarios

Además de la existencia de esta API pública, la aplicación funciona mediante el uso de una API privada, la cual permite acceder a datos almacenados según propiedades privadas como los RFIDs asociados a los objetos o los identificadores privados de cada entidad, ya sea objeto, vídeo o comentario.

Capítulo 5

Implementación

Tras el análisis y el diseño de la aplicación, el siguiente paso es la implementación de la misma. En este apartado se recogen los aspectos más relevantes de esta fase tratando de no incidir en exceso en el código citando, en primer lugar, las tecnologías seleccionadas para el desarrollo principal, así como la estructuración interna de la aplicación y las tecnologías opcionales seleccionadas para facilitar la implementación del proyecto. Por último, se mostrará el aspecto final de la aplicación.

5.1 Tecnologías seleccionadas

Debido a que el presente proyecto será implementado como una aplicación web, las tecnologías que lo conformarán serán decisivas de cara al futuro del mismo, ya que, actualmente, existen diferentes posibilidades a la hora de implementar aplicaciones web, teniendo cada una sus correspondientes ventajas e inconvenientes.

Debido a esto, es necesario elegir una tecnología que no sólo tenga una sencilla implementación y uso, sino que además permita un desarrollo cómodo de la aplicación. Dentro del conjunto de lenguajes de programación comúnmente usados para el desarrollo de aplicaciones web modernas destacan Ruby on Rails, PHP, Python Django, Go Lang y NodeJS.

Elegir cualquiera de estas tecnologías no implicaría ningún inconveniente de cara al desarrollo, ya que todas ellas tienen fuertes similitudes. Por lo tanto, el principal motivo para llevar a cabo una elección correcta será la facilidad de implementación. Teniendo en cuenta esto, la tecnología elegida para este proyecto es NodeJS, ya que el lenguaje usado para el lado del servidor es el mismo que el usado para la capa del cliente: Javascript.

Es importante también señalar que esta tecnología no será la única utilizada para la fase de desarrollo, sino que se hará uso de un conjunto de subsistemas de software existente para el desarrollo de aplicaciones web llamado MEAN Stack [\[15\]](#), cuyas siglas significan:

- **MongoDB**: sistema de bases de datos de la aplicación
- **ExpressJS**: módulo de NodeJS para manejar solicitudes o peticiones HTTP
- **AngularJS**: framework para trabajar en el front end de la aplicación, implementando el patrón MVC (Model View Controller) del lado del cliente
- **NodeJS**: plataforma encargada del funcionamiento del servidor

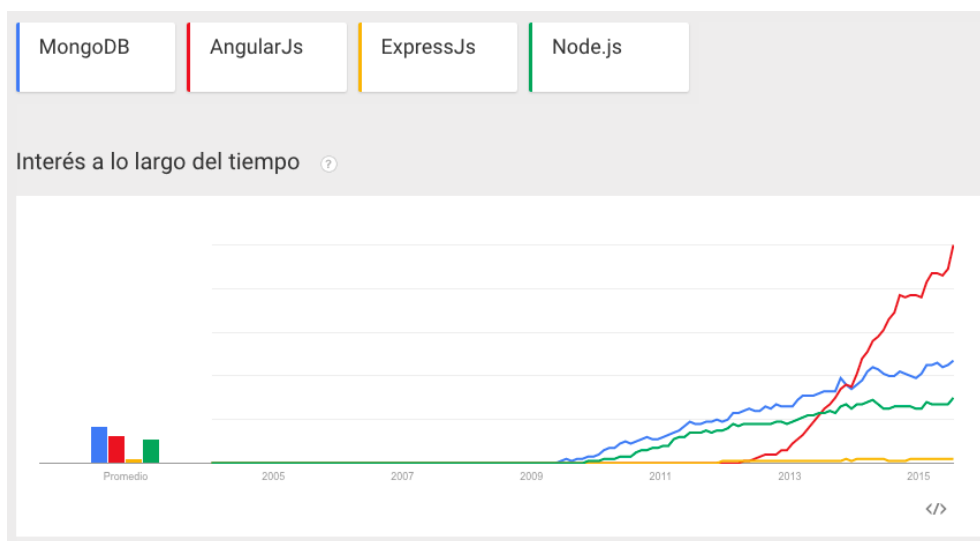


Ilustración 16. Crecimiento de uso e interés de las tecnologías seleccionadas

5.2 Servidor

Como se ha especificado antes, el servidor usará el denominado MEAN stack. Dentro de este conjunto de tecnologías, se incluye el framework ExpressJS, el cual facilita enormemente la modularización y organización del servidor de la aplicación. Con el fin de crear una aplicación escalable, se ha hecho uso de este framework para organizar el servidor siguiendo un patrón similar al anteriormente explicado MVC

(Modelo Vista Controlador). De esta forma, la organización del servidor se compone de los siguientes archivos y carpetas:

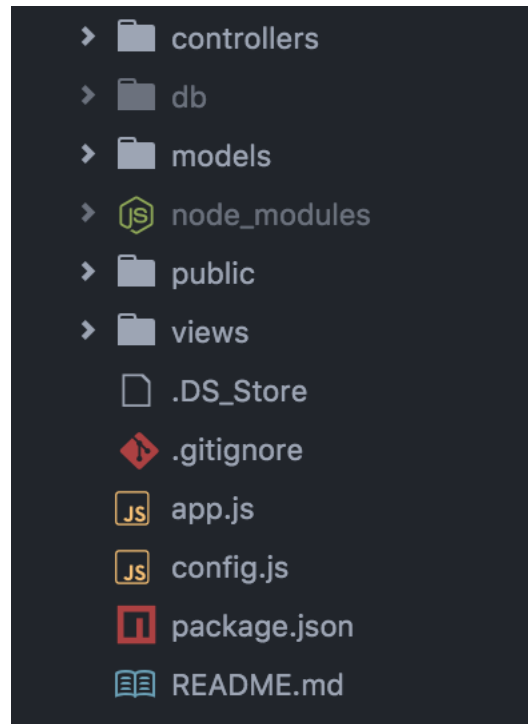


Ilustración 17. Estructura interna del servidor

5.2.1 Carpeta “node_modules”

Esta carpeta contiene todas las dependencias de la aplicación, con el fin de poder importarlas en el archivo principal donde está definido el servidor y hacer uso de las mismas.

5.2.2 Archivo “package.json”

En este archivo se define la información básica de la aplicación, desde el nombre del autor y la versión hasta las dependencias necesarias para el correcto funcionamiento de la misma.

5.2.3 Archivo “app.js”

Este archivo contiene la implementación del servidor, definiendo e inicializando los módulos de NodeJS necesarios para el correcto funcionamiento del mismo. Gracias al uso de ExpressJS, no toda la lógica del lado del servidor estará codificada en este archivo, sino que desde este punto se realizarán llamadas al resto de componentes, con el fin de importarlos y darlos uso.

5.2.4 Archivo “config.js”

En este archivo se almacena información útil y reutilizable por parte del lado del servidor, como puede el puerto asociado al servidor o el directorio de almacenamiento de vídeos.

5.2.5 Carpeta “models”

En esta carpeta se encuentran los archivos que definen cada una de las colecciones existentes en la base de datos. Estos archivos son:

- **Comment.js**, donde se define el modelo a seguir para la colección de comentarios en la base de datos
- **Object.js**, donde se define el modelo a seguir para la colección de objetos en la base de datos
- **Response.js**, donde se define el modelo a seguir para la colección de comentarios respuesta en la base de datos
- **Video.js**, donde se define el modelo a seguir para la colección de vídeos

Todas estas definiciones de modelos para la base de datos se instancian en el archivo “index.js”, el cual, además de recoger todos los modelos existentes, realiza la conexión entre el servidor y la base de datos.

Cabe destacar que, con el fin de mantener una estructura organizada dentro de la base de datos, se utiliza el módulo de NodeJS llamado Mongoose, el cual facilita la definición de modelos para su uso en bases de datos MongoDB y aporta una API que simplifica la conexión y manejo de datos con la base de datos de la aplicación.

5.2.6 Carpeta “views”

En esta carpeta se almacenan las vistas, divididas en módulos, con el fin de renderizarlas en el lado del cliente cuando sea necesario. Para poder tener una organización modular en este apartado, es preciso usar una tecnología diferente a HTML. Existen varias opciones para este fin, entre las que destacan Jade, Marko, Handlebars y EJS. La seleccionada para esta aplicación es Jade, debido al previo conocimiento de esta tecnología por parte del alumno.

5.2.7 Carpeta “controllers”

En esta carpeta se almacenan todos los archivos relacionados con la lógica del lado del servidor. Se realiza una división en tres subcarpetas, con el fin de mantener la estructura modular del servidor:

5.2.7.1 Carpeta “routes”

En esta subcarpeta se almacenan los archivos necesarios para definir las rutas del servidor a las que puede acceder el cliente, con el fin de utilizar la aplicación.

5.2.7.2 Carpeta “api”

En esta subcarpeta se encuentran los archivos que definen la API a ser utilizada por la aplicación. En estos archivos está definida tanto la API pública (aquella disponible para desarrolladores e investigadores) como la API privada (definida para el propio uso de la misma por parte de la aplicación).

Tanto las rutas como la API están basadas en URLs, a las cuales se accede desde fuera del servidor. Aún así, la principal diferencia reside en que el acceso a rutas implica el renderizado de una vista por parte del servidor, para mostrarla en la capa del cliente, mientras que el acceso a la API de la aplicación implica acciones relacionadas con la base de datos, ya sean de inserción o recolección de los mismos.

5.2.7.3 Carpeta “recordrtc”

Con el fin de implementar la subida de vídeos por parte del cliente en tiempo real, fue preciso hacer uso de la biblioteca llamada RecordRTC, la cual permite grabar vídeo y audio en el cliente, y posteriormente subir los datos al servidor, con el fin de almacenarlos.

En esta subcarpeta se encuentran los archivos que definen el comportamiento a seguir por parte del servidor una vez se reciba un vídeo subido por un usuario.

5.2.8 Carpeta “db”

En esta carpeta se almacena la totalidad de la base de datos asociada a la aplicación. Internamente, contiene datos almacenados con extensión BSON (JSON Binario).

5.2.9 Archivo “README.md”

Este archivo Markdown tiene la principal funcionalidad de documentar, de una manera breve y concisa, el propósito del proyecto y su funcionamiento. En este documento se recoge principalmente el diseño de la API pública y la explicación de la puesta en marcha de la aplicación.

5.3 Cliente

Para el correcto funcionamiento de la capa del cliente, únicamente son necesarios las hojas de estilos de la aplicación, junto con la lógica detallada en los manejadores y controladores del cliente (archivos Javascript que definen el funcionamiento y la comunicación con el servidor) y las imágenes que componen la interfaz de usuario.

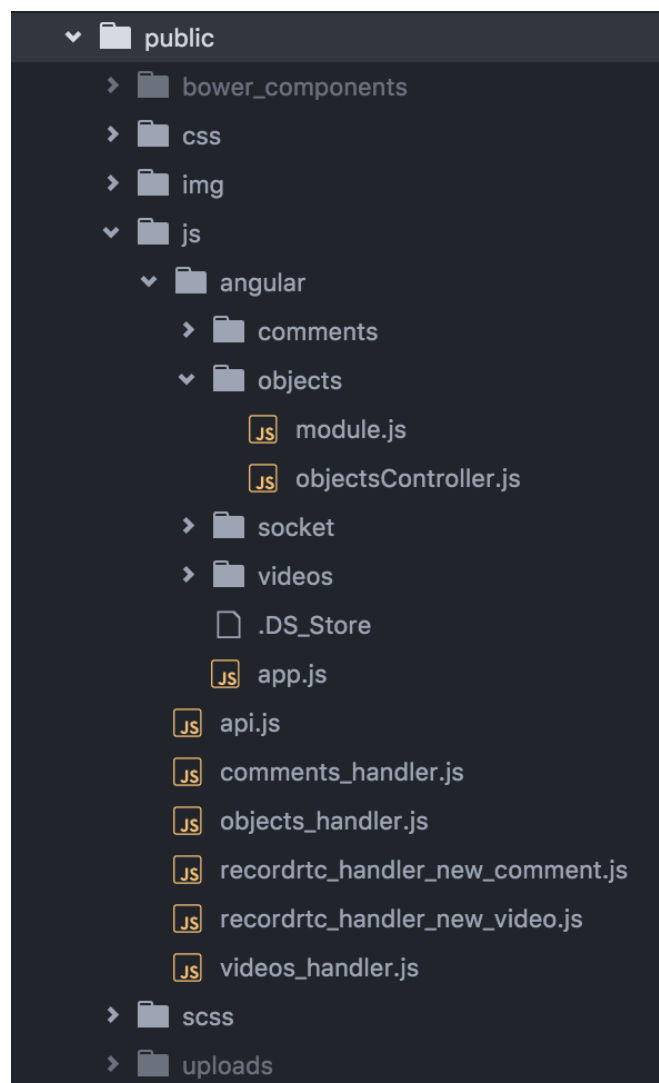


Ilustración 18. Estructura de carpetas de la parte del cliente

En la carpeta “uploads” se almacenarán todos los vídeos subidos por parte de los usuarios, con el fin de que sean de acceso público (ya que, si estuviesen almacenados en las carpetas internas del servidor, no serían accesibles).

Dentro de la carpeta “bower_components” se hallan las dependencias para el correcto funcionamiento de la aplicación en el lado del cliente, entre las que destacan jQuery y RecortRTC.

Por último, es importante señalar que, con el fin de mantener una estructura modular en la capa del cliente, tanto las hojas de estilos como los scripts necesarios están divididos según la funcionalidad que lleven a cabo. De esta manera, la identificación de errores y su consiguiente corrección se puede realizar de manera más rápida.

5.4 Aspecto final de la aplicación

5.4.1 Pantalla principal



Ilustración 19. Aspecto final de la pantalla principal

5.4.2 Pantalla de objeto seleccionado



Ilustración 20. Aspecto final de la pantalla de objeto seleccionado

5.4.3 Pantalla de vídeo seleccionado

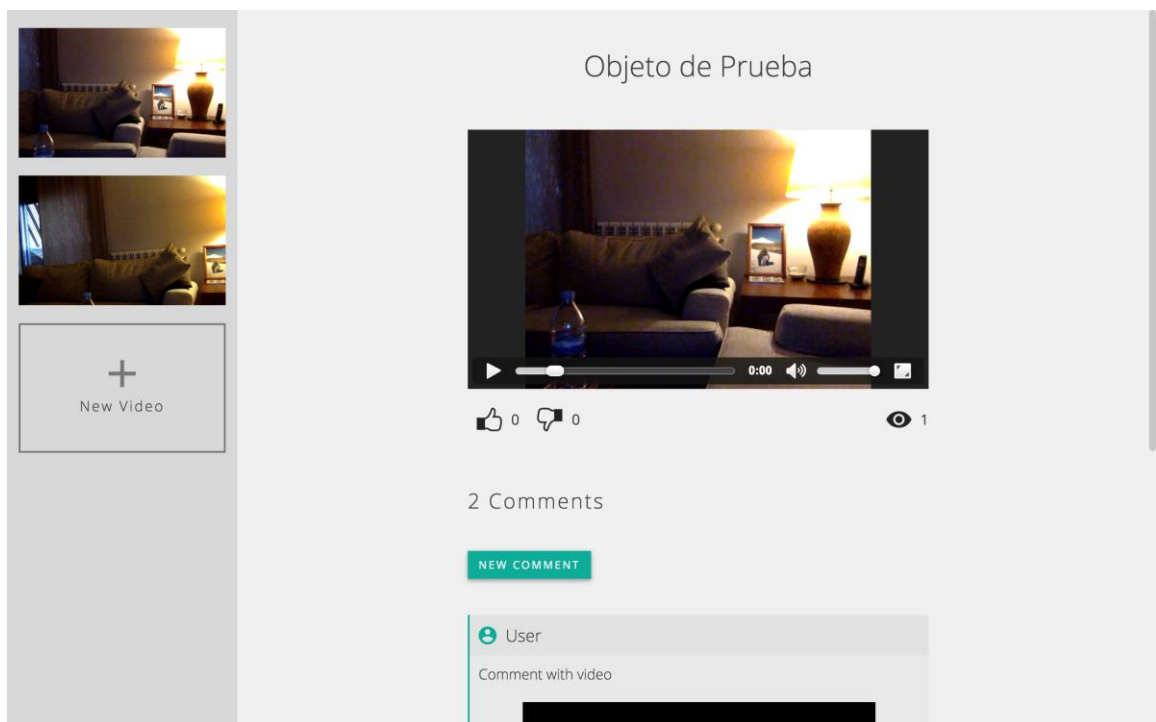


Ilustración 21. Aspecto final de la pantalla de vídeo seleccionado

5.4.4 Pantalla de vídeo seleccionado, sección de comentarios

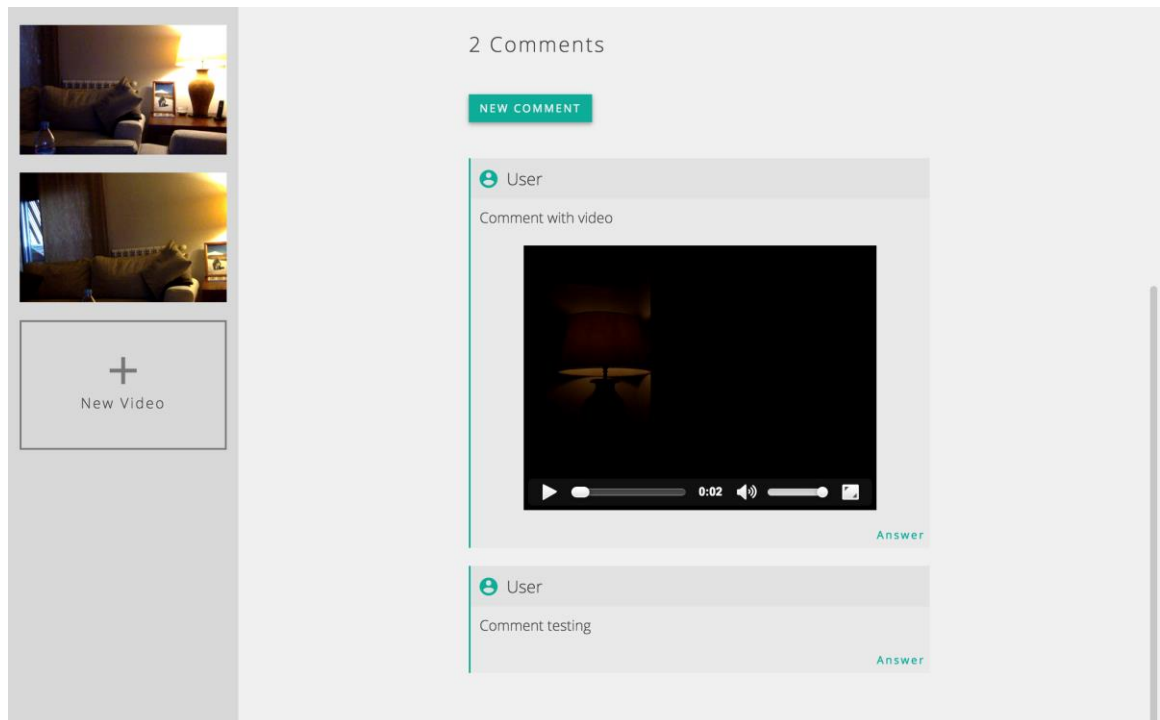


Ilustración 22. Aspecto final de la pantalla de vídeo seleccionado, sección de comentarios

5.4.5 Pantalla de creación de nuevo comentario

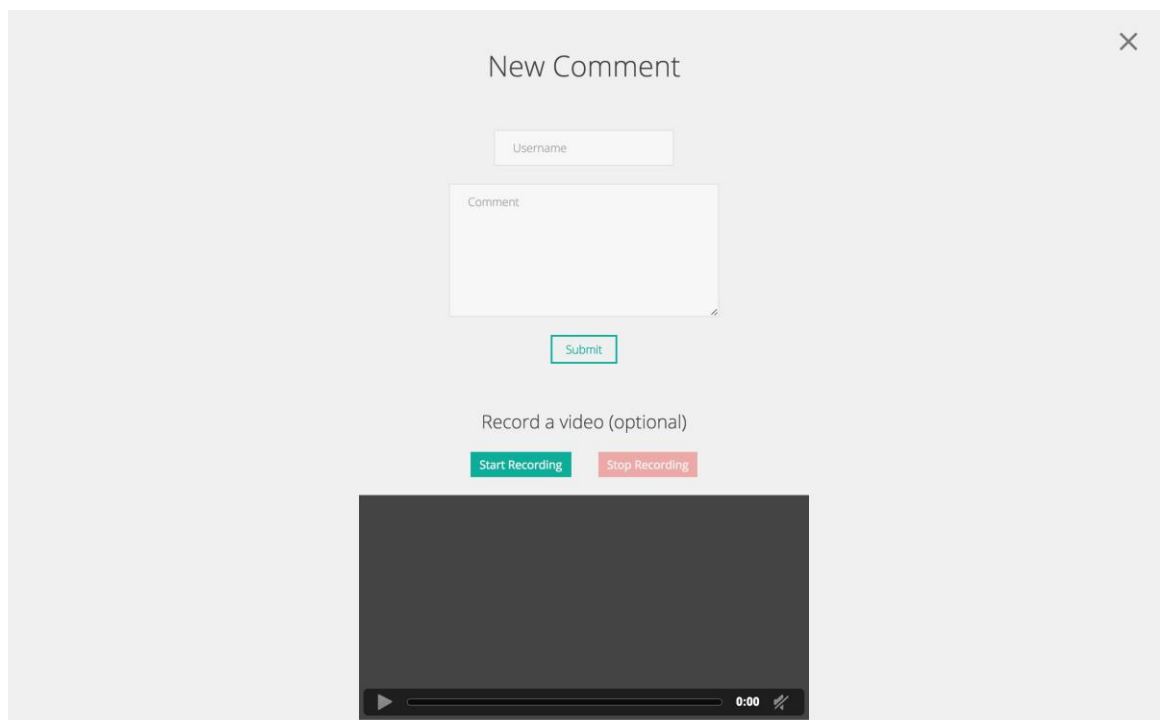


Ilustración 23. Aspecto final de la pantalla de creación de nuevo comentario

5.4.6 Pantalla de creación de nuevo vídeo

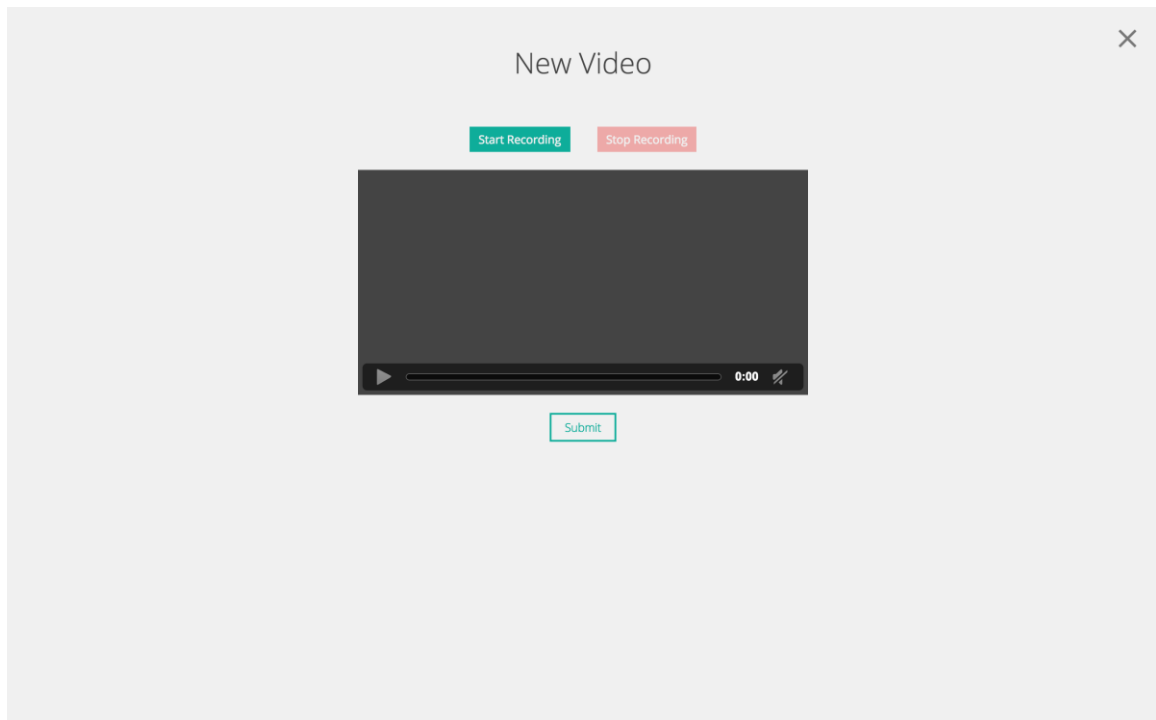


Ilustración 24. Aspecto final de la pantalla de creación de nuevo vídeo

5.4.7 Mensaje de éxito tras crear un comentario

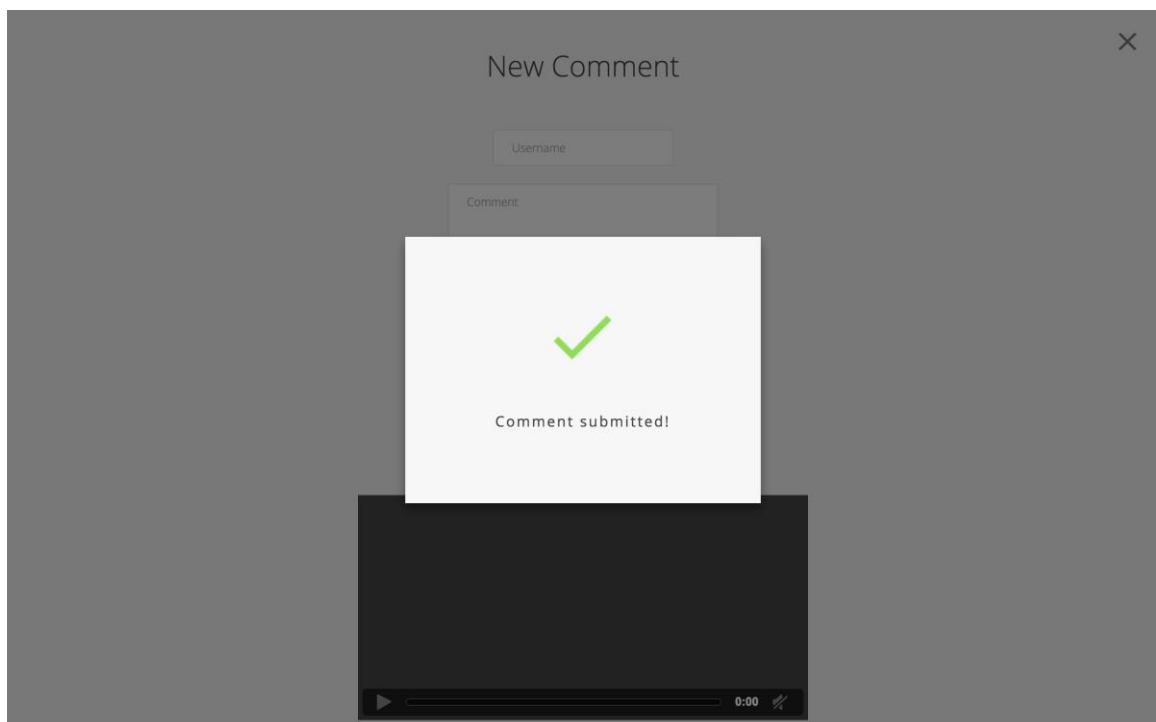


Ilustración 25. Aspecto final del mensaje de éxito tras crear un comentario

5.4.8 Mensaje de éxito tras crear un vídeo

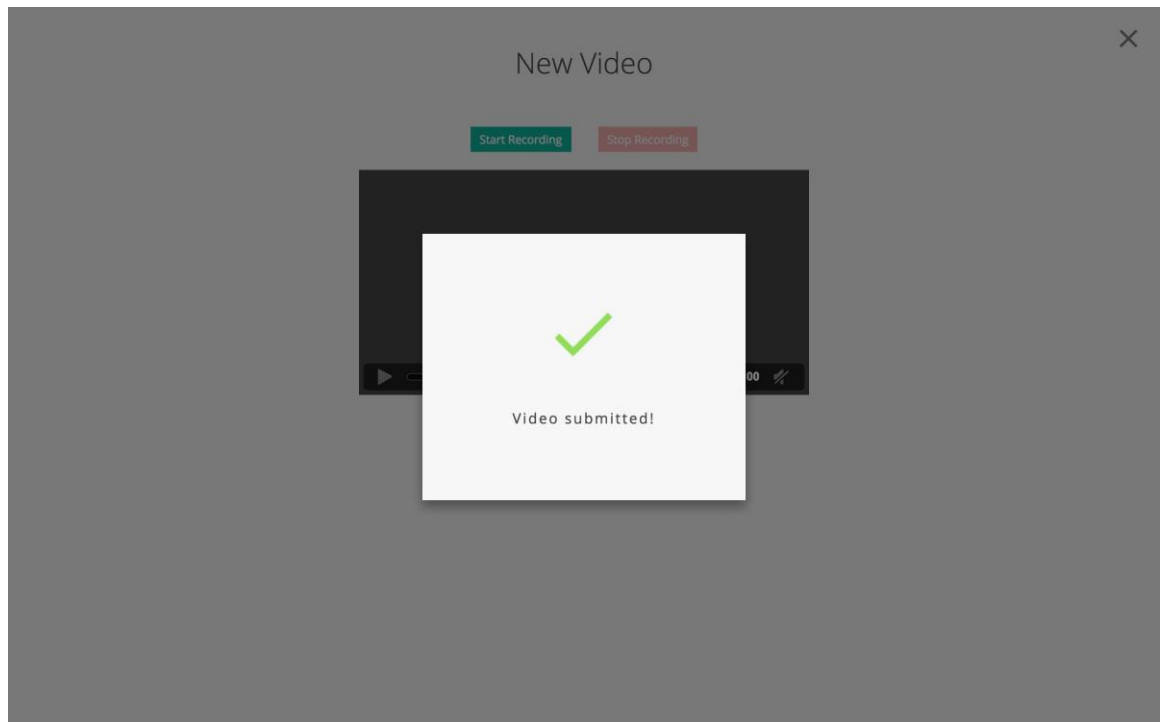


Ilustración 26. Aspecto final del mensaje de éxito tras crear un vídeo

5.5 Pruebas realizadas

Durante el desarrollo de la aplicación, se realizaron diversas pruebas internas para asegurar el correcto funcionamiento de la misma. Sin embargo, estas pruebas no suponían un escenario realista, ya que se basaban en pequeños ensayos de funcionalidad aislados, en las que se sometía a la aplicación a subidas de archivos, recolección de datos y muestra de los mismos, con el fin de encontrar y corregir posibles errores existentes.

Para realizar pruebas más exhaustivas, se llevó a cabo una exhibición de la aplicación en un entorno real con temática cultural, debido a que es la intención principal del proyecto. Esta exhibición se realizó en la Casa del Lector, situada en el centro de creación contemporánea Matadero de Madrid. El objetivo principal de esta demostración fue observar a los usuarios finales mientras usaban la aplicación. Seguidamente, se proporcionaba un test a dichos usuarios, donde se recogían valoraciones personales con el fin de encontrar aspectos a mejorar de la aplicación.

La exhibición tuvo una duración total de 12 horas, distribuidas a lo largo de dos días seguidos. La aplicación fue usada y valorada por más de 35 personas de distintas edades y con distintos tipos de conocimiento sobre la temática del proyecto. Para esta

CAPÍTULO 5: Implementación

demostración, se utilizaron libros de texto de distintos géneros con el fin de atraer a público de distintas edades.



Ilustración 27. Entorno de la exhibición (salón principal de la Casa del Lector, situada en el centro de creación contemporánea Matadero de Madrid)

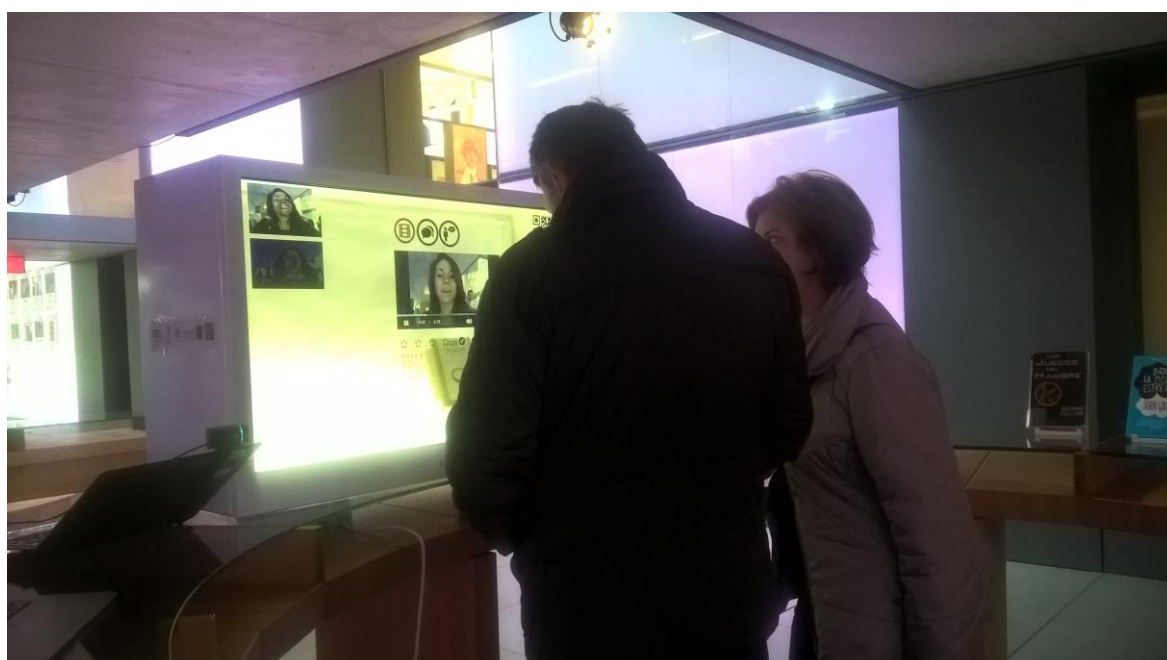


Ilustración 28. Usuarios finales usando la aplicación



Ilustración 29. Escaparate interactivo mostrando la aplicación en ejecución

Las pruebas concluyeron exitosamente, mostrando un gran interés por parte de los usuarios en la tecnología utilizada para el despliegue de la aplicación, al igual que en las posibilidades que brinda este proyecto a eventos de ámbito cultural.

Cabe destacar que, durante la exhibición, la aplicación no mostró errores graves que afectaran a su funcionamiento. A pesar de esto, debido al uso continuo de la misma y también a las pruebas exhaustivas realizadas por parte de los participantes, la aplicación terminó mostrando pequeños defectos en el rendimiento general de la misma, realizando cambios en la interfaz y la base de datos de forma más lenta de lo habitual.

5.6 Integración con plataformas culturales

Tras finalizar el desarrollo de la aplicación, junto con su correspondiente API pública, y haber realizado pruebas con usuarios finales, esta se desplegó en un servidor público, con el fin de permitir el uso de la API de recolección de datos. De esta forma, la información recolectada durante las pruebas realizadas y futuras exhibiciones podrá ser accedida de forma pública.

Es importante señalar que el contenido almacenado en la aplicación es de utilidad para el proyecto europeo meSch [\[16\]](#), cuyo objetivo es co-diseñar nuevas plataformas para la creación de exposiciones tangibles en el patrimonio cultural. Este proyecto utiliza distintas colecciones de datos, provenientes de distintos proyectos creados por diferentes universidades e instituciones.

Es importante señalar que uno de los objetivos principales de meSch es la carencia de los elementos materiales para poder acceder a la información que estos nos aportan, dejando que el apartado físico continúe en exposiciones.

La principal diferencia que aporta este proyecto a las colecciones de datos disponibles en el proyecto europeo meSch es el crecimiento de la propia colección gracias al aporte de los usuarios con su uso. La mayoría de colecciones actualmente disponibles en meSch proviene de la digitalización de obras existentes en museos. Debido a esto, la información aportada por las distintas instituciones colaboradoras en el proyecto tiende a ser estática, donde los cambios posibles en los datos estarán sujetos a cambios en dichas obras.

En cambio, la colección de información aportada por este proyecto está basada principalmente en los datos generados por el uso de la aplicación. De esta forma se fomenta la comunicación e interacción entre usuarios de eventos culturales, con el fin de compartir sus experiencias. Es interesante contemplar este punto debido al carácter social que pretende aportar dentro del ámbito cultural.

Para poder acceder a los datos almacenados en la base de datos de este proyecto, será preciso utilizar la plataforma virtual de meSch (www.mesch.io), desde la cual se podrán realizar búsquedas en cualquier colección de datos disponible.

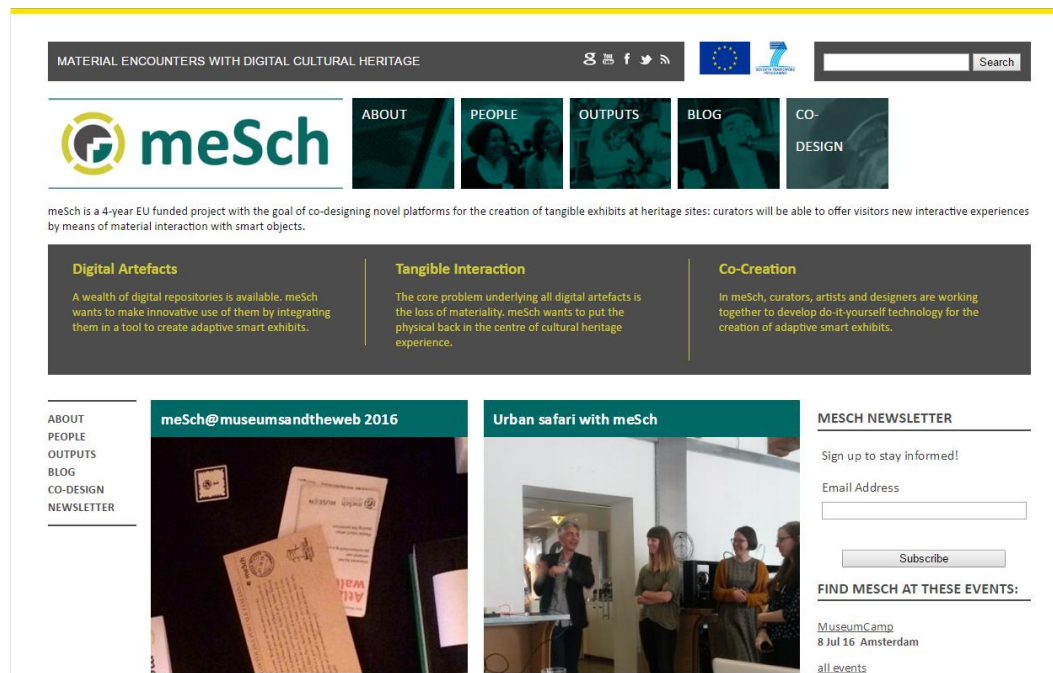


Ilustración 30. Pantalla principal del proyecto meSch (mesch-project.eu)

Search for content



Here you can search for content items that can be used as draft version for your content items. After adding the content items your form will be filled with the necessary content. Feel free to edit the page.

Search for description, titles, creators, age	Europeana
---	-----------

recipe content > [search](#)

Media

- ☒ Image
- ☒ Text
- ☒ Sounds
- ☒ Video

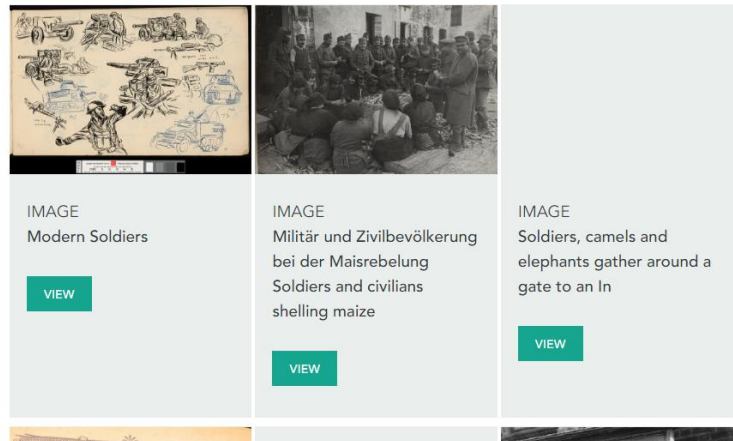


Ilustración 31. Pantalla de búsqueda del proyecto meSch

Capítulo 6

Gestión del proyecto

Para el correcto desarrollo de un proyecto, es pertinente conocer las fases que lo componen, así como el orden de ejecución de las mismas. Esta información se representa a través del modelo de ciclo de vida elegido. Además, cada fase se divide en un conjunto de actividades o tareas que son planificadas previamente para garantizar la consecución de los objetivos en el tiempo disponible. Por último, estas actividades tienen asociados unos costes de personal y recursos que se calculan elaborando un presupuesto.

6.1 Modelo de ciclo de vida del software

Como se ha mencionado, la elaboración de un proyecto se divide en un conjunto de fases y actividades que se desarrollan con un orden específico. Esto se halla definido mediante el denominado modelo de ciclo de vida del software. Para el desarrollo de este proyecto, el modelo del ciclo de vida elegido es el modelo en cascada en su variante realimentada.

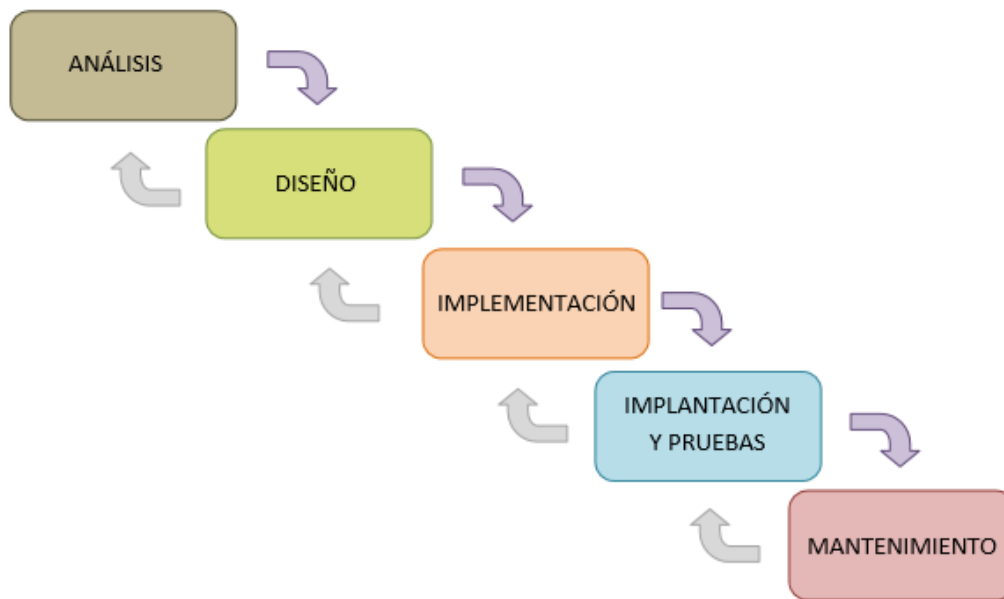


Ilustración 32. Ciclo de vida en cascada

Como puede observarse en la ilustración anterior (ilustración 16), las fases se llevan a cabo secuencialmente, ya que la salida de una fase compone la entrada de la siguiente. Se trata de una variante realimentada dado que, en caso de existir algún fallo en una fase avanzada, puede regresarse a fases anteriores y realizar los cambios necesarios en esa y las siguientes fases.

El ciclo de vida en cascada es el más sencillo de poner en práctica y, aunque en algunos proyectos de gran envergadura no sea el más apropiado, las condiciones de este proyecto lo hacen una buena opción dado que se trata de un sistema pequeño, perfectamente definido desde su comienzo y en el que se esperan muy pocos o nulos cambios.

6.2 Planificación

Mediante la planificación del proyecto es posible obtener una aproximación del tiempo necesario para su desarrollo. Para ello, se establecen las actividades y tareas que componen su elaboración y se estima el tiempo necesario para realizar cada una de ellas. Una buena herramienta para realizar esta tarea son los diagramas de Gantt. Para la elaboración del mismo, se supone una jornada laboral media de 5 horas, lo que supone 35 horas de trabajo semanal.

El diagrama correspondiente se muestra en la siguiente ilustración. Tal y como se aprecia en la figura, el proyecto finalmente tuvo una duración de 100 días, comenzando el día 16 de febrero de 2016 y terminando el día 25 de mayo de 2016.

6.2 Planificación

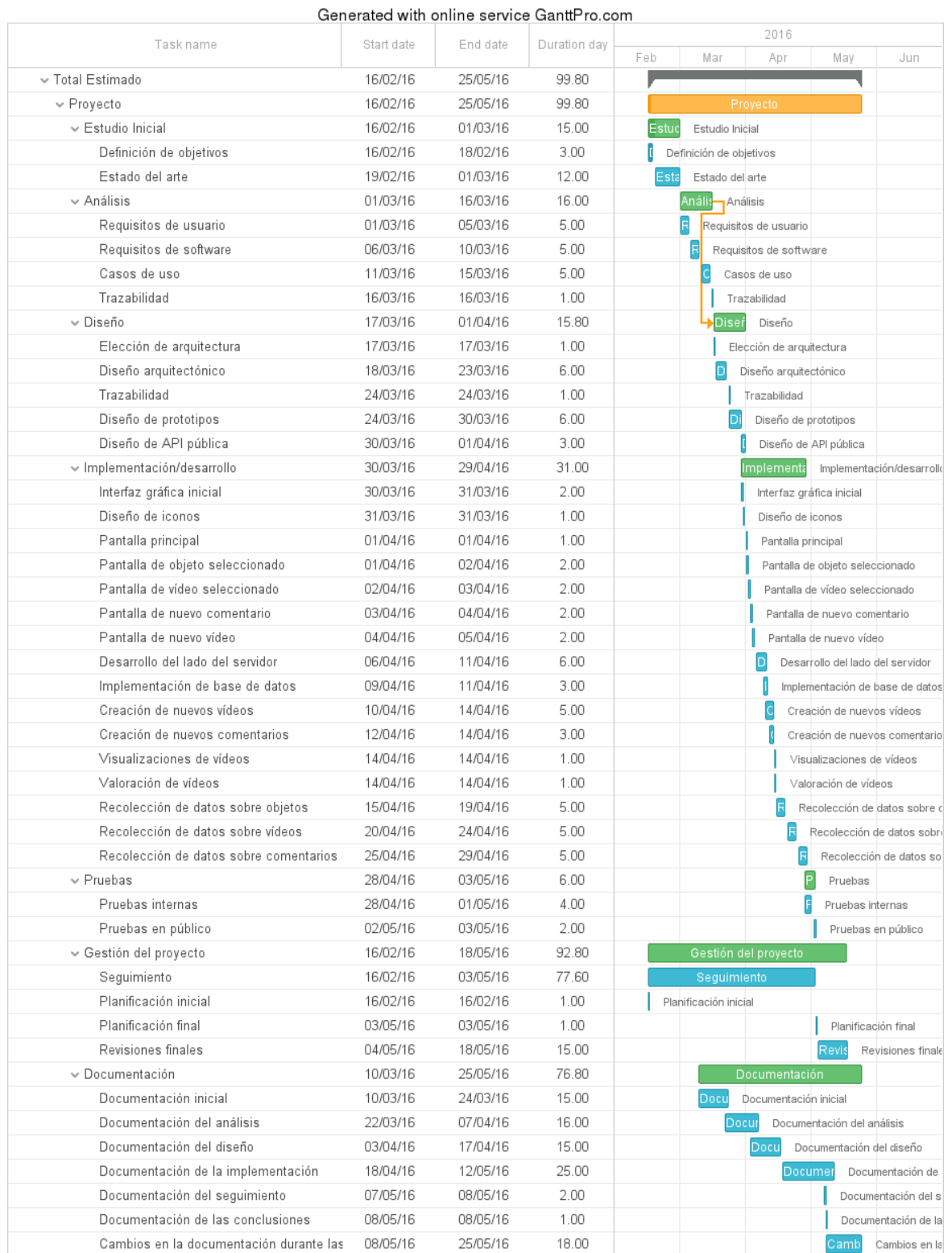


Ilustración 33. Diagrama de Gantt

6.3 Presupuesto

Para llevar a cabo las tareas detalladas anteriormente, se necesita personal y material, lo que deriva los costes que se detallan a continuación.

6.3.1 Personal

Teniendo en cuenta la situación de desarrollo del proyecto, solo existen dos personas a cargo del mismo. Por ello, todos los roles necesarios para su elaboración son repartidos y asumidos por el tutor y el alumno. De esta manera, el tutor adopta el rol de cliente y el alumno hace las veces de jefe de proyecto, analista, diseñador y programador según corresponda.

Como se ha obtenido en la planificación del apartado anterior, el número total de días dedicado al desarrollo del proyecto asciende a 100. En cada jornada de trabajo se invierte una media de 5 horas, lo que deriva en un total de 500 horas de dedicación por parte del alumno.

A continuación, se muestra un resumen de los costes totales de personal, derivados de la realización del proyecto.

COSTES DE PERSONAL			
Rol	Horas dedicadas	Coste por hora (€/hora)	Coste total (€)
Jefe de proyecto	75	50	3.750
Diseñador	80	40	3.200
Analista	100	30	3.000
Programador	245	20	4.900
		TOTAL	14.850 euros

Tabla 75. Costes de personal

En conclusión, el gasto total de personal asciende a CATORCE MIL OCHOCIENTOS CINCUENTA EUROS (14.850€), IVA no incluido.

6.3.2 Material

El coste asociado al material necesario para el desarrollo del proyecto se detalla a continuación.

COSTES DE MATERIAL						
Descripción	Coste (€)	Uds.	% de uso dedicado	Periodo de depreciación (meses)	Uso (meses)	Total (€)
iMac 21.5"	1.529	1	80	36	3	101,93
Macbook pro 13"	1.649	1	100	36	3	137,41
Samsung Galaxy Tab 4 10.1"	199	1	30	18	2	66,3
Escaparate interactivo MyMultitouch	5.000	1	70	36	2	194,44
					TOTAL	500,08

Tabla 76. Costes de material

La fórmula de amortización utilizada es la siguiente:

$$\frac{\text{Nº de meses de utilización}}{\text{Período de depreciación}} \times \text{Coste del equipo} \times \% \text{ de uso dedicado}$$

Ilustración 34. Fórmula de amortización

6.3.3 Resumen de costes

El coste total de la elaboración del proyecto se resume a continuación.

RESUMEN DE COSTES	
Concepto	Coste total (€)
Personal	14.850,00
Material	500,08
TOTAL	15.350,08

Tabla 77. Resumen de costes

Finalmente, puede concluirse que el coste total del proyecto asciende a QUINCE MIL TRESCIENTOS CINCUENTA EUROS CON OCHO CÉNTIMOS (15.350,08 €), IVA no incluido.

Capítulo 7

Conclusiones y trabajo futuro

Para concluir el documento, se realiza una síntesis del objeto y desarrollo de la aplicación, así como una propuesta de posibles líneas futuras de mejora de la misma.

7.1 Conclusiones

Los continuos cambios en nuestra sociedad llevan hacia la necesidad de potenciar y modernizar aspectos relacionados con el patrimonio cultural. Sin embargo, para impulsar este cambio es vital proveer de herramientas que permitan brindar a los usuarios un nuevo modo de interacción con las obras, con el fin de generar contenido basado en sus experiencias, al igual que permitir el acceso a dicho contenido para su uso en futuras aplicaciones e investigaciones.

Como medio para llevar a cabo esta tarea se han utilizado tecnologías web, basándose en la arquitectura Modelo Vista Controlador y en el stack MEAN para el desarrollo de aplicaciones web, la cual usa como lenguaje principal Javascript.

Puesto que la principal meta era crear una aplicación fácil de usar, fue de gran importancia realizar un correcto análisis de requisitos, teniendo en cuenta las necesidades de los usuarios finales. También, con el mismo propósito, se buscó realizar un cuidado diseño de la aplicación, especialmente en lo relacionado a la interfaz gráfica.

Para ello se realizaron prototipos de bajo nivel que sirvieron de guía a la implementación y que pretendían conseguir pantallas visuales e intuitivas.

Además, en este proyecto, se ha podido comprobar cómo se implementan actualmente catálogos y exposiciones virtuales de eventos culturales, y se han detallado sus ventajas e inconvenientes. En este sentido puede ser interesante la creación de estándares que permitan su fácil compresión, acceso y uso por parte de usuarios finales y desarrolladores.

Atendiendo a la utilidad de la aplicación desarrollada, es interesante observar la interoperabilidad entre proyectos basados en tecnologías web. Un claro ejemplo de esto es el anteriormente explicado proyecto europeo meSch, en el cual se ha añadido una referencia al presente proyecto, de forma que es posible realizar búsquedas de datos sobre la información recolectada en la base de datos de la aplicación.

En el ámbito en el que se encuentra este proyecto, es interesante observar que es un primer paso para la creación de narrativas alternativas para las obras presentes en eventos culturales, basadas en los propios usuarios de dichos eventos, aportando contenido variado y dinámico. En las pruebas realizadas se ha podido observar que los visitantes encuentran esta herramienta de utilidad y muestran interés en su funcionamiento.

Por último, en el apartado personal, este proyecto ha servido para repasar y afianzar muchos de los conceptos y conocimientos adquiridos a lo largo de la carrera. También, ha requerido el aprendizaje de estándares y buenas prácticas en el desarrollo de aplicaciones web, a la vez que superar distintos problemas en el camino, resultando ser una experiencia enriquecedora y gratificante.

7.2 Líneas futuras de mejora

Si bien la aplicación final cumple con el objetivo principal para el que fue pensada, es cierto que existe margen para la mejora de la misma, introduciendo nuevas funcionalidades.

La primera de ellas sería la adaptación de la propia aplicación a dispositivos móviles, permitiendo reutilizar la capa de servidor, pero prescindiendo de usar un navegador para pasar a usar de forma nativa el dispositivo móvil.

Una segunda línea de mejora estaría relacionada directamente con la recolección de información almacenada en la base de datos, añadiendo más características a los datos generados con el fin de poder realizar búsquedas más complejas y exhaustivas.

Por último, también sería de interés mejorar la compatibilidad entre navegadores web, teniendo en cuenta los formatos de multimedia utilizados en la generación de datos que puedan causar problemas en este ámbito.

Capítulo 8

Referencias

- [1] Web Apps are the future. Disponible en Internet:
<http://www.sitepoint.com/long-live-web-app>. Accedido en marzo de 2016.
- [2] Duggan, M., Ellison, N. B., Lampe, C., Lenhart, A., & Madden, M. (2015). Social media update 2014. *Pew Research Center*, 19.
- [3] Vieweg, S., Hughes, A. L., Starbird, K., & Palen, L. (2010, April). Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 1079-1088). ACM.
- [4] Bellucci, A., Diaz, P., & Aedo, I. (2015, November). A See-Through Display for Interactive Museum Showcases. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces* (pp. 301-306). ACM.
- [5] Open API. Disponible en Internet.
https://en.wikipedia.org/wiki/Open_API. Accedido en marzo de 2016.
- [6] RFID. Disponible en Internet:
<https://es.wikipedia.org/wiki/RFID>. Accedido en marzo de 2016.
- [7] Web Application. Disponible en Internet:
https://en.wikipedia.org/wiki/Web_application. Accedido en abril de 2016.

[8] Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1), 210-230.

[9] SixDegrees.com. Disponible en Internet:
<https://en.wikipedia.org/wiki/SixDegrees.com>. Accedido en abril de 2016.

[10] Twitter. Disponible en Internet:
<https://es.wikipedia.org/wiki/Twitter>. Accedido en abril de 2016.

[11] Schweibenz, W. (2004). Virtual museums. *The Development of Virtual Museums*, „ICOM News Magazine”, (3 s 3).

[12] Interfaz de programación de aplicaciones. Disponible en Internet
https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones.
Accedido en abril de 2016.

[13] Representational State Transfer. Disponible en Internet:
https://es.wikipedia.org/wiki/Representational_State_Transfer. Accedido en abril de 2016.

[14] Mashup. Disponible en Internet:
[https://en.wikipedia.org/wiki/Mashup_\(web_application_hybrid\)](https://en.wikipedia.org/wiki/Mashup_(web_application_hybrid)). Accedido en junio de 2016.

[15] MEAN. Disponible en Internet:
<https://es.wikipedia.org/wiki/MEAN>. Accedido en abril de 2016.

[16] Mesch Project. Disponible en Internet:
<http://mesch-project.eu>. Accedido en junio de 2016.

Anexo

Project Summary

Abstract

Nowadays, technology has an important role in our daily life, allowing us to, for example, communicate with other people around the world and share our thoughts by just using small devices connected to the Internet. This way of expressing ourselves has been gaining more importance since the creation of social media, where people with similar interests or with real-life connections are directly connected to each other. One of the most clear examples of this is Twitter, where people express themselves by just using 140 characters. Also, it is now common for cultural events such as museums to use technology in order to give their users a richer experience. This technology usage can go from the existence of audio guides and the usage of multimedia for showing extra information of different pieces of art to the availability of virtual exhibitions.

This project aims to provide a way of letting the users share their experiences and opinions when attending a cultural-related event. As an example, users could rate, make a comment and even create a video talking about a painting inside a museum, and also be able to view other people's opinions about that same object. Moreover, this information could be used by developers by taking advantage of the application public API, in order to retrieve the stored data generated by the own users and use it for future research.

The different stages of the application design and development are described in this document, starting from the collection of the prerequisites and concluding with the final implementation and deployment.

Introduction and objectives

Nowadays, Internet is a crucial element regarding applications, being almost an essential requirement for these to work in mobile, web and desktop environments. Because of this, web applications development has been growing in the last decade, since these are totally based in a web browser and its capacities, easing the network integration inside these apps.

Regarding web applications, it can be said that social media is particularly important, due to the large number of active users they have, which use these applications daily, generating huge amounts of information. This information can be useful in different situations outside the application. An example of this is the use of data stored on Twitter, where the information provided by users in real time can be used to track an emergency situation.

Given this versatility of web applications and the information generation led by it, this project's main objective is the development of an application for events of a cultural nature, based on a similar social networking approach, providing users with a way to interact with other visitors of these events. The information generated by using this application will be accessible to developers and researchers, enabling future research and applications based on it.

Regarding the technology currently used in cultural events, this is focused on virtual tours and museum catalogs, which consist of a virtual representation of the content offered by the event. These tours and catalogs are commonly accessible through the Internet, which brings both advantages and disadvantages, which will be explained later.

Currently, the curators of cultural property decide which information will be presented in an exhibition. This project proposes to enable visitors to participate actively in creating alternative narratives to the ones proposed by curators.

The main motivation of this project stems from the lack of tools that allow users to interact with each other in an event in which all share a common factor. It is interesting to note that technology in this area is currently used for another approach, improving the experience that the event itself can provide to these users, but not encouraging the communication between them. Also, this project also presents a tool for developers, allowing public access to the data generated by users. Thus, the use of the application to develop will imply a growth of the same, providing a vast and rich virtual catalog with varied content.

The decision to use web technologies in this project lies in the ease they provide when developing a multiplatform application and also in the integration with the Internet, because of the intention to allow access to the information stored.

Objectives

This project focuses on the study and development of a prototype web application that gives users of cultural events the opportunity to express, communicate and evaluate opinions of other users of the same event.

Taking into account the above explained, it is intended to design a web application specially aimed at two types of different users:

- End-users: those using the application in order to interact with the artwork available in the museum (share experiences related to these works, rate and comment on them) and other users (through communication based on video and text). As a result, the use of the application by these users involve data generation and its subsequent storage.
- Developers: which access the data stored by the application in order to use it in projects outside the current one's scope.

Because the project will have the two aforementioned types of users, it is important to note that the interaction with the application will be different in both cases:

- For end-users, the application will be used by an interactive display and an RFID code reader, which will facilitate interaction with the chosen work, while providing some degree of immersion.



Ilustración 35. Interactive showcase for which the application is designed, including the RFID code reader inside

- For developers, stored data of the application may be accessed from any environment that supports the HTTP protocol. For doing this, a public API will be designed and implemented for communicating with the server layer application from an external environment.

Document content

This document is divided into several chapters, where all the information related to the development of the project is collected. These chapters are:

- Introduction: an overview of the project will be given, which objectives have been established and what the content of the rest of the document will be.
- State of the art: some generalities of web technologies and how they have been crucial in the development of applications in recent years, and how technology and the cultural sphere have joined in order to enrich the experience of the users.
- Analysis: the functionality of the application will be covered, taking into account the functionality that the user expects from it. Later, use cases will be detailed, where the operations that users can perform using the application will be clearly seen.

- Design: the system context and application architecture will be explained, as well as the traceability matrix of the project.
- Implementation and development: here, the technologies used in the development of the application will be discussed.
- Project Management: in this chapter, it is described and justified the life cycle model chosen for the project. Then, the planning as well as the costs associated with the development of the Project will be presented.
- Conclusions: In this chapter, the main results of the project are summarized and possible lines for future improvement are offered.
- References: In this chapter, bibliographic resources consulted during the preparation of the project are listed.

Testing

During the development of the application, various internal tests were performed to ensure the proper operation of it. However, these tests did not pose a realistic scenario, as they were based on small isolated functionality tests, in which the file uploading, data collection and visualization of the data was tested, in order to find and correct any existing errors.

In order to perform more extensive tests, an exhibition of the application in a real cultural environment was carried out, because it is the main intention of the project. This exhibition was held at the Casa del Lector, located in the center of contemporary creation Matadero Madrid. The main objective of this demonstration was to observe end users while using the application. Next, a test was given to those users, where personal ratings were collected in order to find ways to improve the application.

The exhibition had a total duration of 12 hours spread over two days. The application was used and evaluated by more than 35 people of different ages and with different kinds of knowledge on the subject of the project. For this demonstration, textbooks of different genres were used, in order to attract audiences of different ages.



Ilustración 36. Exhibition environment (main room of Casa del Lector, located in the center of contemporary creation Matadero de Madrid)

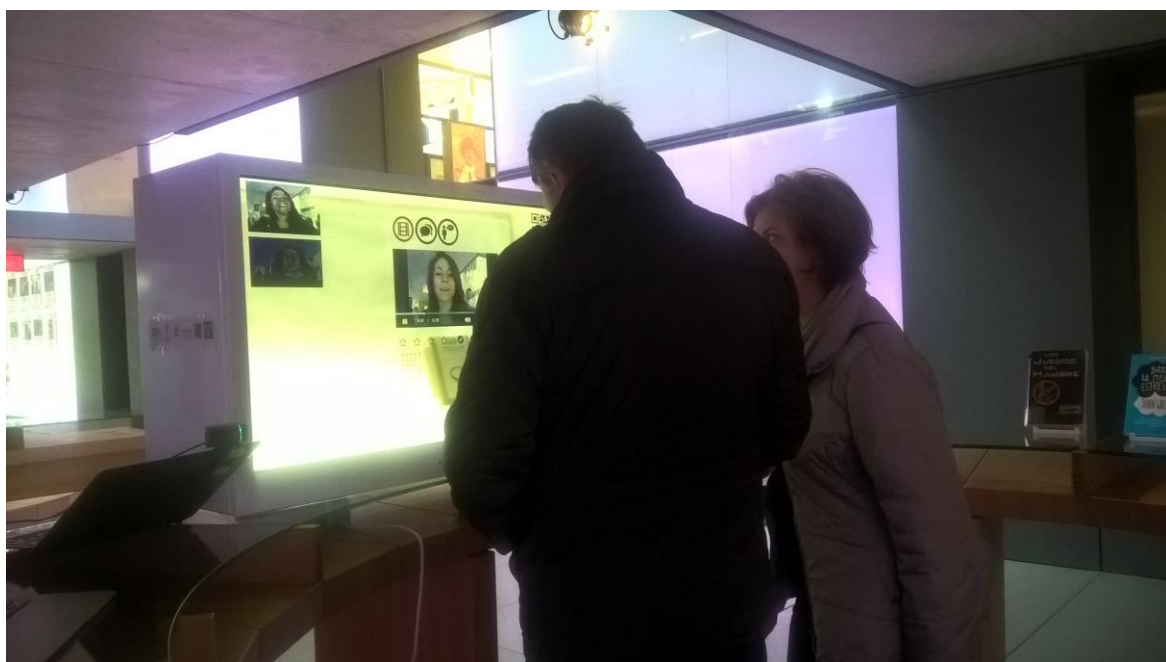


Ilustración 37. End-users using the application



Ilustración 38. Interactive showcase showing the application

The tests were successfully concluded, showing great interest from users in the technology used to deploy the application, as in the possibilities offered by this project to events cultural field.

Note that the application did not show serious errors affecting its operation during the exhibition. Despite this, due to the continuous use of it and also to extensive testing by participants, the application ended up showing small defects in the overall performance of the same, making changes to the interface and database way slower than usual.

Integrating with cultural platforms

After completing the development of the application, along with its corresponding public API for collecting data, and having tested it with end-users, the project was deployed on a public server, in order to allow the use of the data collection API. Thus, the information collected during tests and future exhibitions could be accessed publicly.

It is important to note that the content stored in the application is useful for meSch European project, which aims to co-design new platforms for creating tangible cultural heritage exhibitions. This project uses different collections of data from different projects created by different universities and institutions.

It is important to note that one of the main objectives of meSch is the lack of material elements to access the information they give us, letting the physical section continue in exhibitions.

The main difference that this project provides data collections available in meSch European project is the growth of the collection itself thanks to the contribution of users with their use. Most currently available meSch collections comes from the digitization of existing works in museums. Because of this, the information provided by the various collaborating institutions in the project tends to be static, where any data changes depend on the changes in the real works.

Instead, the collection of information provided by this project is based primarily on data generated by the use of the application. Thus communication and interaction between users of cultural events, in order to share their experiences, is encouraged. It is interesting to contemplate this point because of the social character which aims to bring within the cultural field.

For accessing the data stored in the database of this project, it will be necessary to use the virtual platform Mesch (www.mesch.io), from which you can search any collection of data available.

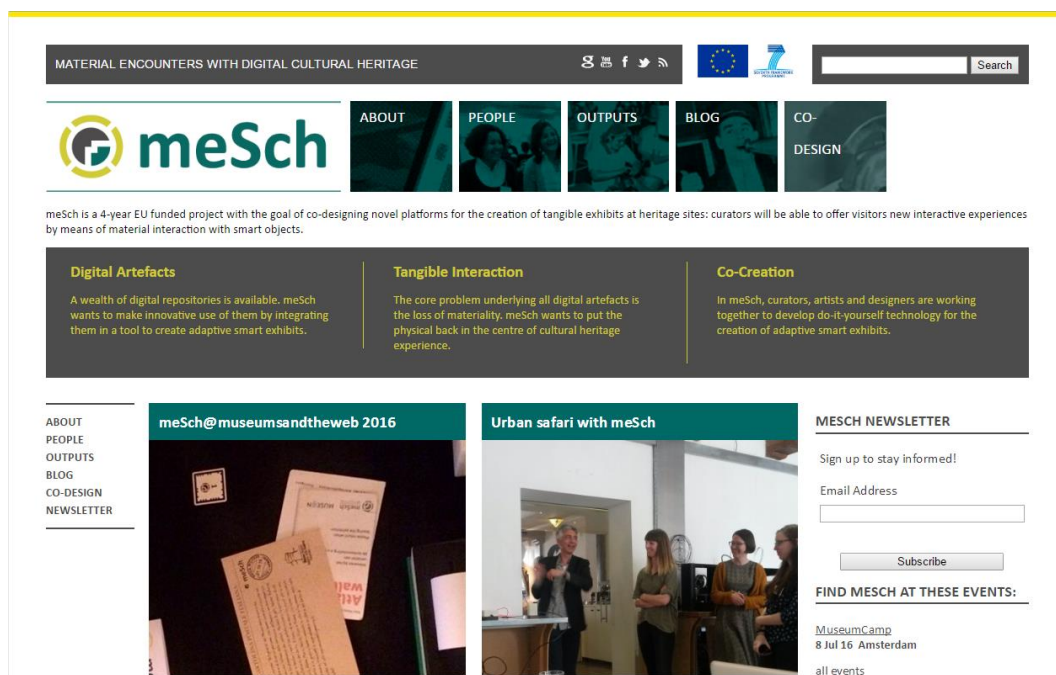


Ilustración 39. Main Screen of meSch project (mesch-project.eu)

Search for content

Here you can search for content items that can be used as draft version for your content items. After adding the content items your form will be filled with the necessary content. Feel free to edit the page.

[recipe content](#) > [search](#)

Media

- ☒ Image
- ☒ Text
- ☒ Sounds
- ☒ Video

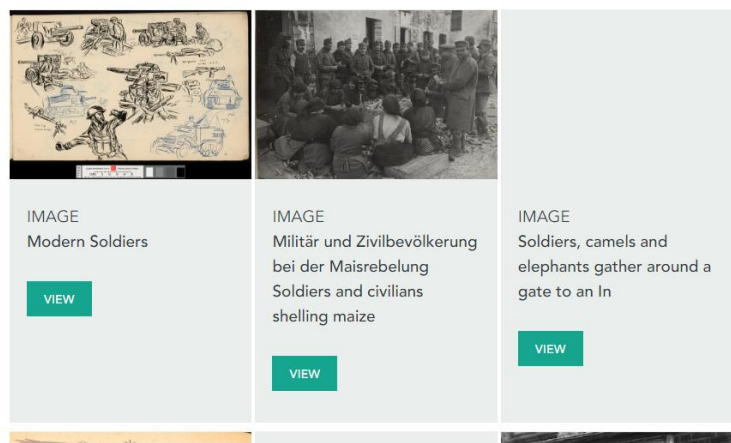


Ilustración 40. Search screen within meSch project

Conclusions

The continuous changes in our society lead to the need to strengthen and modernize aspects of cultural heritage. However, to drive this change is vital to provide tools to give users a new way of interaction with the artworks, in order to generate content based on their experiences, as well as allow access to such content for use in future applications and research.

In order to accomplish this task, several web technologies have been selected and used, based on the Model View Controller architecture and the MEAN stack for web application development, which uses Javascript as its main language.

Since the main goal was to create an easy to use application, it was of great importance to do a proper requirements analysis, taking into account the needs of end users. Also, for the same purpose, it was intended to make a careful application design, especially related to the graphical interface. For accomplishing this, low-level prototypes that served as guide to implementation were created, in order to get visual and intuitive screens.

Furthermore, in this project, it has been seen how current catalogs and virtual exhibitions of cultural events are implemented, detailing their advantages and disadvantages. In this sense it can be interesting to create standards that allow easy compression, access and use by end users and developers.

Taking into account the usefulness of the application developed, it is interesting to observe the interoperability between projects based on web technologies. A clear example of this is the European project Mesch explained above, which recently added a reference to this project, so that it is possible to search for data on the information collected in the database of the application.

In the field in which this project resides, it is interesting to note that it is a first step in order to create alternative narratives for the works in cultural events, based on the users of these events, providing varied and dynamic content. In the tests to which this project has been subjected, it has been observed that visitors find this tool useful and show interest in its functioning.

Finally, in the personal section, this project has served to revise and consolidate many of the concepts and knowledge acquired throughout the degree. It has also required learning standards and best practices in the development of web applications, while overcoming various problems along the way, proving to be an enriching and rewarding experience.

Future improvement areas

While the final application meets the main objective for which it was intended, it is true that there is scope for improving it, introducing new features.

The first one would be the adaptation of the application itself to mobile devices, enabling reuse the server layer, but putting aside the need for a browser and using the mobile device natively.

A second improvement would be directly related to the collection of information stored in the database, adding more features to the generated data in order to perform more complex and exhaustive searches.

Finally, it would also be interesting to improve the compatibility between web browsers, considering multimedia formats used in the generation of data that can cause problems in this area.

